Learning and Leveraging Structured Knowledge from User-Generated Social Media Data

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy by

Hang Dong

April 2020

Contents

List of Figures

List of Tables

Notations

The following key notations are found throughout this thesis:

C	Finite set of tag concepts
C_a	A tag concept a
C_b	A tag concept b
$\mathbb{F}$	Folksonomy
$\mathbb{F}^{clean}$	Cleaned Folksonomy
$\mathbb{F}^{str}$	Structured Folksonomy
N	Number of occurrence of all tag concepts
N_z	Number of occurrence of all tag concepts assigned to topic z
R	Finite set of resources
$R_{a,b}$	Common root (or parent) tag concept of C_a and C_b
T	Finite set of tags
U	Finite set of users
V	Finite set of vocabularies
$\|V\|$	Vocabulary size
X	The collection of instances (each as a sequence)
$\|X\|$	Instance (or document) size
Y	Set of all possible labels
$\|Y\|$	Label size
$\mathbf{z}$	The set of all hidden topics
$\|\mathbf{z}\|$	Number of hidden topics
$\mathbf{z}_a^{sig}$	The set of all significant topics for the concept C_a
$\|\mathbf{z}_a^{sig}\|$	Number of significant topics for the concept C_a
z	A hidden topic
Sim	Label similarity matrix
Sub	Label subsumption matrix
W_*	Weight matrices in neural networks (Subscript * matches to any characters)
$\overrightarrow{Y_i}$	The multi-hot vector representation of the label set for the ith instance
b_*	Bias vectors in neural networks (Subscript * matches to any characters)

$\mathbf{c}_a$	The original content representation vector
$\mathbf{c}_i$	The document representation vector
$\mathbf{c}_t$	The title-guided content representation vector
$\mathbf{c}_{ta}$	The title representation vector
d_e	Dimensionality of neural word embeddings
$h^{(t)}$	The hidden state vector at the time step t
$\tilde{h}^{(t)}$	The candidate hidden state vector at the time step t
$\mathrm{n_a}$	Number of words in the content
$\mathrm{n_s}$	Number of sentences in the content
$\mathrm{n_t}$	Number of words in the title
$r^{(t)}$	Reset gate vector at the time step t
s_{ij}	Value of the jth node in the output layer given the ith instance as input
t	A time step in a Recurrent Neural Network
$v(C_a)$	The topic distribution vector of the tag concept a
v_{sa}	The attention vector in the original sentence-level attention mechanism
v_{wa}	The attention vector in the word-level attention mechanism for the content
v_{wt}	The attention vector in the word-level attention mechanism for the title
x	An instance as a sequence
x_t	Sequence of words in the title
x_a	Sequence of words in the content (or abstract)
y_{ij}	Binary value indicating relevancy of the jth label to the ith instance
$z^{(t)}$	The update gate vector at the time step t
$\alpha^{(i)}$	Attention score of the ith word in the title
$\alpha_s^{(r)}$	Attention score of the rth sentence in the content
λ_1	The regularisation parameter for L_{sim}
λ_2	The regularisation parameter for L_{sub}
L	The whole loss function
L_{CE}	The binary cross entropy loss
L_{sim}	Similarity loss
L_{sub}	Subsumption loss

Abbreviations

The following key abbreviations are found throughout this thesis:

AdaBoost	Adaptive Boosting
AI	Artificial Intelligence
Bi-GRU	Bi-directional Gated Recurrent Units
CART	Classification and Regression Trees
CCS	ACM Computing Classification System
CNN	Convolutional Neural Networks
DAG	Directed Acyclic Graph
GRUs	Gated Recurrent Units
HAN	Hierarchical Attention Network
JMAN	Joint Multi-label Attention Network
KB	Knowledge Base
KG	Knowledge Graph
KL Divergence	Kullback-Leibler Divergence
KOSs	Knowledge Organisation Systems
LDA	Latent Dirichlet Allocation
LR	Logistic Regression
LSTM	Long-Short Term Memory
MCG	Microsoft Concept Graph
Q&A	Question and Answering
RBF	Radial basis function
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
TF	Taxonomic F-measure
TP	Taxonomic Precision
TR	Taxonomic Recall

Preface

This thesis is primarily my own work. The sources of other materials are identifed.

Abstract

Knowledge has long been a crucial element in Artificial Intelligence (AI), which can be traced back to knowledge-based systems, or expert systems, in the 1960s. Knowledge provides contexts to facilitate machine understanding and improves the explainability and performance of many semantic-based applications. The acquisition of knowledge is, however, a complex step, normally requiring much effort and time from domain experts. In machine learning as one key domain of AI, the learning and leveraging of structured knowledge, such as ontologies and knowledge graphs, have become popular in recent years with the advent of massive user-generated social media data.

The main hypothesis in this thesis is therefore that a substantial amount of useful knowledge can be derived from user-generated social media data. A popular, common type of social media data is social tagging data, accumulated from users' tagging in social media platforms. Social tagging data exhibit unstructured characteristics, including noisiness, flatness, sparsity, incompleteness, which prevent their efficient knowledge discovery and usage. The aim of this thesis is thus to learn useful structured knowledge from social media data regarding these unstructured characteristics. Several research questions have then been formulated related to the hypothesis and the research challenges.

A knowledge-centred view has been considered throughout this thesis: knowledge bridges the gap between massive user-generated data to semantic-based applications. The study first reviews concepts related to structured knowledge, then focuses on two main parts, *learning structured knowledge* and *leveraging structured knowledge* from social tagging data. To learn structured knowledge, a machine learning system is proposed to predict subsumption relations from social tags. The main idea is to learn to predict accurate relations with features, generated with probabilistic topic modelling and founded on a formal set of assumptions on deriving subsumption relations. Tag concept hierarchies can then be organised to enrich existing Knowledge Bases (KBs), such as DBpedia and ACM Computing Classification Systems. The study presents relation-level evaluation, ontology-level evaluation, and the novel, Knowledge Base Enrichment based evaluation, and shows that the proposed approach can generate high quality and meaningful hierarchies to enrich existing KBs. To leverage structured knowledge of tags, the research focuses on the task of automated social annotation and propose a knowledge-enhanced deep learning model. Semantic-based loss regularisation has been proposed

to enhance the deep learning model with the similarity and subsumption relations between tags. Besides, a novel, guided attention mechanism, has been proposed to mimic the users' behaviour of reading the title before digesting the content for annotation. The integrated model, Joint Multi-label Attention Network (JMAN), significantly outperformed the state-of-the-art, popular baseline methods, with consistent performance gain of the semantic-based loss regularisers on several deep learning models, on four real-world datasets.

With the careful treatment of the unstructured characteristics and with the novel probabilistic and neural network based approaches, useful knowledge can be learned from user-generated social media data and leveraged to support semantic-based applications. This validates the hypothesis of the research and addresses the research questions. Future studies are considered to explore methods to efficiently learn and leverage other various types of structured knowledge and to extend current approaches to other user-generated data.

Acknowledgements

This thesis would not have been completed without the support and guidance of my supervisors. First and foremost, I would like to express my deepest gratitude to my main supervisor, Dr Wei Wang, for his valuable advice, knowledge, and insights, as well as his constant support and patience to my research progress over the years. Wei encourages me to improve my research, to never stop thinking, and to become a maturer researcher who explores his potential in different aspects. It has been a great honor to have the opportunity to work with him.

I would also like to thank my second supervisors, Prof Kaizhu Huang and Prof Frans Coenen. It has been my privilege to join the research group with other colleagues mainly led by Kaizhu, who encourages me to achieve higher objective in research. Also, during the time in the University of Liverpool, Frans discussed with me about research regularly and kindly introduced me to the Natural Language Processing Group.

Many thanks go to all my colleagues, to name a few, Wei Wang, Xianbin Hong, Shiyang Yan, Jie Zhang, Zhiqiang Bi, Yuji Dong, Qi Chen, etc., who have supported me during the years as friends and partners in the department.

Finally, and most importantly, I would like to deeply thank all my family members, especially my father and mother, who always supports me during the time of my PhD study and ever since I was born. Thanks for your persistent, unconditional love, caring, and understanding. I would never be able to complete this PhD without you. Forever thanks for the enduring, loving words and wishes my mom gave me, and to my dad for your positive attitude towards life that always influences me. The time with you, when we are together or on the phone, is always the most precious for me.

Chapter 1

Introduction

Knowledge is love and light and vision. – Helen Keller [96, p. 20]

Although language is a human construct, that does not make it transparent to us. Like the children we make, the meanings we make can have secrets from us. – Timothy Williamson [195]

Knowledge has long been a crucial element throughout the history of Artificial Intelligence (AI). In 1956, the first type of structured knowledge was created as an interlingua (or an artificial, intermediary language) by Richard H. Richens of the Cambridge Language Research Unit to support machine translation [106]. From the 1960s, the prevailing knowledge-based systems or expert systems started to rely mainly on domain knowledge to build applications in AI. An early, representative knowledge-based system was DENDRAL [113], developed in the chemical domain and inspired later applications.

For semantic-based applications such as text mining, information retrieval, and recommendation, knowledge provides contexts to facilitate language understanding, and thus improves the performance and the explainability of the applications. For more than a half-century, researchers have been interested in knowledge representation [148, p. 468-p. 473]: creating large Knowledge Bases (KBs) to represent and store knowledge as relations, hierarchies, and formal ontologies, from manual creation to knowledge extraction from data, covering specific and general domains. Some of the representative projects include Cyc[1] [107] and DBpedia[2] [17]. There have already been formal and vivid discussions on leveraging structured knowledge (or "ontologies") in machine learning since the 2000s, especially for text mining applications [21]. A more recent, well-known example of knowledge acquisition is the Google Knowledge Graph project [50, 155], employed in 2012 to enhance the information retrieval of things, people, and places on the Web.

[1] `https://www.cyc.com`
[2] `https://wiki.dbpedia.org/`

While the acquisition of knowledge is a complex step requiring tremendous effort and time from domain experts, the advent of Web 2.0 and recent progress in machine learning provide new perspectives for knowledge acquisition. Web 2.0 is a social Web: users have become producers of content, rather than just consumers. The present Web contains a massive amount of data and content created by users by every second, especially on social media platforms. This big volume of data enables us to explore the "wisdom of the crowd". Besides, recent advances in machine learning allow better learning and leveraging of strictured knowledge. Probabilistic topic models can infer meanings from unstructured texts in an unsupervised way [18, 19, 190]. After the resurgence of neural networks and deep learning, the representation and reasoning of knowledge in deep learning applications become a research frontier to explore [68, p. 482].

A popular, common type of social media data is social tagging data, accumulated from users' tagging in social media platforms. It is however that, just like other types of social media data, social tagging data suffer inherent problems as human languages, such as noisiness and ambiguity. The complex meanings of social tags are challenging for computational processing. The set of user-generated tags is also flat and have low semantics, which prevents efficient organisation, browsing, search and semantic-based recommendation of online content. Sparsity is also a problem of tagging data due to many unique tags and little contextual information. Finally, social tagging data are incomplete and many shared resources are not associated with any tags. All these issues prevent efficient utilisation of the collection of these user-generated social media data.

Taking into account knowledge as a crucial and central element in AI and the issues of user-generated social media data, the thesis aims at learning structured knowledge from social media data, especially from social tagging data. The learning process and the learned structured knowledge will address the issues of noisiness, ambiguity, flatness of tagging data. Moreover, structured knowledge has to be useful for downstream semantic-based application, thus the thesis also concerns leveraging structured knowledge from tagging data for automated social annotation. This task of automated social annotation will further tackle the issues of incompleteness of tagging data.

The rest of this introduction chapter is organised as follows. The background and motivation of this research is presented in Section 1.1. Then the aim and the scope of this thesis is discussed in Section 1.2, followed by the general and the specific research questions in Section 1.3, regarding both learning structured knowledge and leveraging structured knowledge. Research contributions regarding the two aspects are presented in Section 1.4. The overview of this thesis is outlined in Section 1.5 with a short summary for each chapter.

1.1 Background and Motivation

As stated at the beginning, from the early AI systems to the recent progress in machine learning, knowledge plays key roles as contextual backgrounds to improve the

performance and explainability of many semantic-based applications. The acquisition of knowledge, however, requires a tremendous amount of effort by domain experts and knowledge engineers. Since the advent of Web 2.0, many studies have been focusing on deriving knowledge from user-generated social media data or the "collective intelligence", contributed by the massive number of users on the Web.

A common and popular type of data in social media platforms is crowdsourced from social tagging. Since the creation of the social tagging system Delicious[3] in 2003, and Flickr[4] in 2004, tagging has become a built-in functionality in many social media sites, users can share and annotate resources with their own vocabularies. In academic social bookmarking systems, such as Bibsonomy[5] and CiteULike[6], tags are annotated to organise academic papers; in social question & answering (Q&A) sites, such as Quora[7], StackOverFlow[8] and Zhihu[9], tags are associated to questions for better search and recommendation; in microblogging services like Twitter[10] and Weibo[11], tags are in the form of hashtags to produce alternative access points to microblogs. There are also social media platforms supporting tagging of images, such as Flickr and Instagram[12]; tagging of movies, such as MovieLens[13] and Douban Movie[14]; and tagging of musics, such as last.fm[15] and Xiami[16], etc. Figure 1.1 displays the user interface of the Bibsonomy website, including resources (bookmarks and papers), tags, users, etc.[17].

These accumulated tags form "folksonomies" (a portmanteau of "folk" and "taxonomies") [184], which are perceived as valuable user-generated metadata to supplement controlled vocabularies for resource organisation [116, 215], users' browsing [126], information retrieval and recommendation [61, 133]. It is also discovered that tags have higher descriptive and discriminative powers compared to other textual features, such as titles, descriptions and comments, for document classification [54].

Folksonomies represent a key technology since the age of Web 2.0, where users have become producers rather than just consumers of content on the Web. Massive amount of data, especially on social media platforms, are created and shared by users every second[18]. These fast-accumulated data, however, due to their unstructured, noisy and ambiguous nature, are challenging to be processed to acquire "collective intelligence". Thus, there has been a consensus in the research communities to derive *structured knowledge*

[3] `https://del.icio.us/`
[4] `https://www.flickr.com`
[5] `https://www.bibsonomy.org`
[6] `http://www.citeulike.org`
[7] `https://www.quora.com`
[8] `https://stackoverflow.com`
[9] `https://www.zhihu.com/`
[10] `https://twitter.com`
[11] `https://weibo.com`
[12] `https://www.instagram.com`
[13] `https://movielens.org/`
[14] `https://movie.douban.com`
[15] `https://www.last.fm`
[16] `https://www.xiami.com`
[17] For details of this interface, see `https://www.bibsonomy.org/help_en/User_Interface`.
[18] Statistics in `https://www.internetlivestats.com/one-second/`.

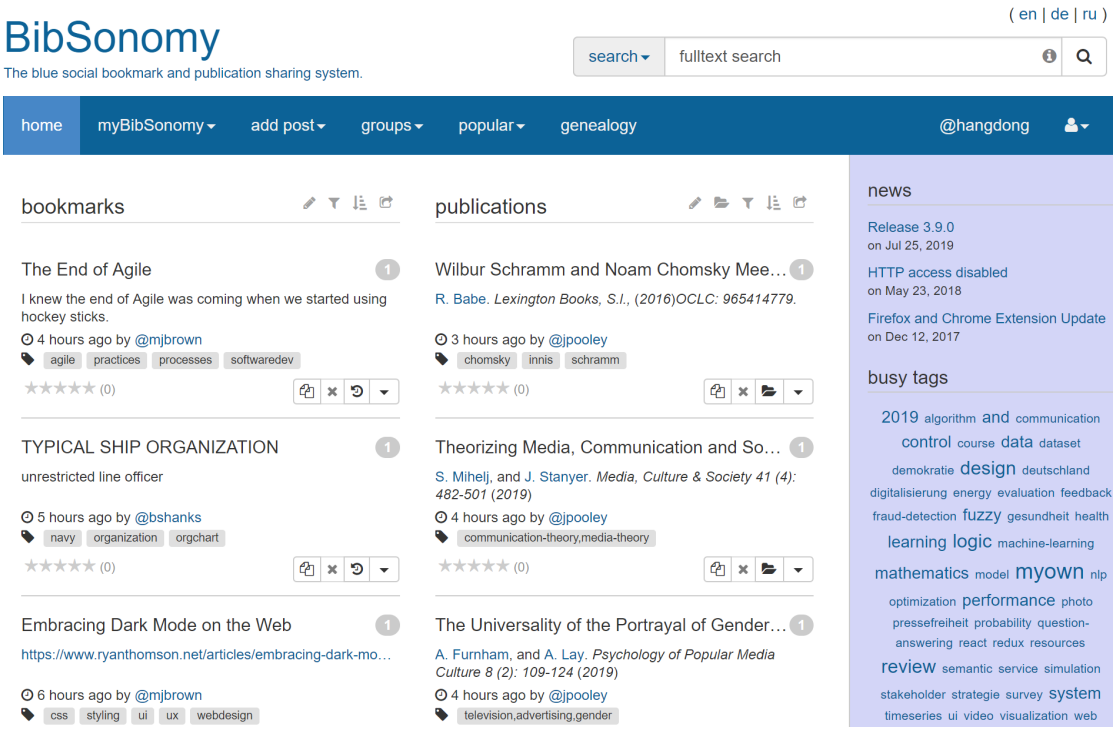


FIGURE 1.1: The user interface of the Bibsonomy website (screencast in August 2019), including resources (bookmarks, on the left column, and publications, on the right column), tags (marked with grey background), users (marked with @ sign). The bottom right part shows the "busy tags", or the currently most popular tags.

from social media data, including social tagging data, to support many semantic-based applications.

To learn structured knowledge from social tagging data is challenging due to inherent problems of social tagging data, similar in human language, elaborated in [135, p.218-228]. Several key challenging issues are highlighted and addressed in this thesis. The first problem, that hinders the usage of tagging data as a knowledge source, is the lack of controlled vocabulary. This results in noisiness and ambiguity of tagging data. Social tags, freely contributed by different users, have various morphological variations and ambiguous meanings [58, 90]. This motivates the research for data cleaning and concept extraction from noisy social tags.

Second, social tags are inherently flat and lack a structured form. Unlike the traditional classification systems, the set of tags does not define their relations [135, p. 222-p. 223]. This prevents the use of tags to efficiently support semantic-based navigation, information retrieval, and recommendation. Thus many studies have attempted to explore methods to infer structured knowledge, such as relations, concept hierarchies, and lightweight ontologies, from tags [58, 124, 165]. Limitations of the current methods motivate to learn more accurate and useful structured knowledge from social tagging data [48].

Another problem with social tagging data is sparsity. This is most common for social tagging data in the academic domain (such as Bibsonomy and CiteULike), having

slower acculumation of tags [51] and lower agreement between users [51] than the social tagging data in more general domains (such as Delicious). Also, unlike other forms of user-generated texts such as online comments and (micro-)blogs, social tags contain little contextual information. As such, popular and effective methods, e.g., Hearst lexico-syntactic patterns [78] for mining subsumption relations from documents and Web pages [196] cannot be adopted directly.

The other issue is the incompleteness of tagging data. Many shared documents in social media platform are not associated with any tags. For example, in the social question and answering (Q&A) website Zhihu, more than 18% of questions are not associated with any tags, as reported in [130]. In the microblogging website Twitter, only 10% to 15% of tweets are associated with at least one hashtag [97, 191]. This is mainly because annotating objects requires much cognitive effort and can be time-consuming. Also as just discussed, some of the user-generated tags are noisy and of low quality. All of these make the tags not as useful as expected. It is believed in this thesis that these problems can be alleviated to a great extent by automated annotation, which learns from the cleaned user-generated tagging data to suggest a set of tags for previously unseen documents.

Thus in general, this thesis attempts to address the above challenges in user-generated social media data, with methods in natural language processing, data mining, and machine learning. Previous studies aimed at exploring tag semantics simply based on data co-occurrence, which is sensitive to data sparsity [80] and cannot explicitly and formally define the relations among tag [58]. With recent advances in probabilistic topic modelling [18, 19, 190], it is necessary to revisit the challenging task of learning structured knowledge from social media data. There is also a lack of formal evaluation studies to assess the quality of the learned knowledge, especially on the newly enriched knowledge not contained in existing KBs.

To tackle the needs for automated annotation of socially shared documents, recent studies adapt deep learning models and formalise the task as multi-label classification [67, 77, 89, 110, 210]. Multi-label classification typically needs to take the relations among labels into consideration. This is actually the structured knowledge derived from the tagging data in the context of social text annotation. Although there are recently many studies on representing and utilising knowledge to enhance neural network models to improve performance and explainability [24, 25, 101], little research focused on leveraging structured knowledge of labels for deep learning based multi-label classification.

1.2 Aim and Scope of the Study

Based on the research motivations above, the aim of this study is thus to learn useful structured knowledge from social media data. The study focuses on data from social tagging, which is a common functionality across many social media platforms. More specifically, the study mainly explores academic social tagging data, i.e., tagging data

for academic publications (such as Bibsonomy and CiteULike), as structured knowledge that can be derived from academic resources is of particular interest to the research community. The task is also more challenging than learning in the general domain as the academic domain contains sparser data [51, 80, 183].

Knowledge is crucial to AI and machine learning. The learned structured knowledge has to be applied to support downstream semantic-based applications. As a natural extension over the part of *learning structured knowledge*, the second part of the thesis focuses on a key semantic-based application, automated social text annotation, that requires *leveraging structured knowledge* from the tags. The task of automated social text annotation predicts tags from the input documents, that mimics the user tagging process. Recent studies mostly use deep learning approaches to model the annotation process and formulate the task as a multi-label classification problem. Thus, the study attempts to address the label correlation issue in deep learning based multi-label classification, through leveraging structured knowledge.

1.3 Research Questions

The main hypothesis in this research is that *a substantial amount of useful knowledge can be learned from user-generated social media data.* The study focuses on social tagging data as a typical and common type of user-generated social media data. Based on this hypothesis, the research in this thesis explores several questions. The first question, that also reflects the research aim, is about learning structured knowledge from social tagging data. This covers several specific questions regarding the unstructured characteristics of social tagging data, the approaches to learn and to evaluate structured knowledge. Corresponding to the extension from learning structured knowledge to leveraging structured knowledge, the other main question is about leveraging structured knowledge in the semantic-based, machine learning application, automated social annotation. This question involves two aspects: (i) the use of knowledge in machine learning for multi-label classification, and (ii) the modelling of users' tagging process. The two main questions are thus:

- *How to learn structured knowledge from user-generated social media data?*
- *How to leverage structured knowledge in machine learning to support automated social text annotation?*

These questions can be split into more specific questions, corresponding to the issues and challenges identified from the tasks:

- Q1: *How to address the noisiness, ambiguity, sparsity, and incompleteness issues of social tagging data?*
- Q2: *How to learn subsumption relations and concept hierarchies from social tagging data?*

- Q3: *How to formally evaluate the learned structured knowledge from social tagging data?*
- Q4: *How to leverage structured knowledge to tackle the label correlation issue in deep learning based multi-label classification?*
- Q5: *How to model users' social annotation process through deep learning?*

Questions 1 to 3 are relevant to the learning of structured knowledge, regarding the inherent unstructured characteristics from the user-generated social tagging data (Q1), the general methods to infer structured knowledge (Q2), and the evaluation of the learned structured knowledge from social media data (Q3).

Questions 4 to 5 are relevant to leveraging structured knowledge, for the semantic-based application, automated social annotation, which will help address the incompleteness issue of social tagging data. The key question (Q4) is to explore the novel approaches to utilise structured knowledge to address the label correlation issue in multi-label classification. Based on recent advances of deep learning methods, the last question (Q5) asks to adapt methods and techniques in deep learning to efficiently model the users' annotation process.

1.4 Research Contributions

To answer the research questions above, the thesis provides contributions as follows, organised in terms of learning structured knowledge and leveraging structured knowledge.

The thesis, first, reviewed the relevant concepts of *structured knowledge* and identified the key aspects from the literature related to both learning and leveraging structured knowledge from social tagging data. Then, in terms of learning structured knowledge,

- A supervised machine learning system is designed to learn subsumption relations from academic social tagging data. The machine learning system takes a novel perspective of the semantics of user-generated tags, where a tag is viewed as a complex entity that potentially has different meanings under different contexts or subject areas. After a data cleaning module to address the noisiness and a part of sparsity issues of tagging data, the systems resorts to probabilistic topic modelling to represent each tag concept as a distribution of latent topics. With this representation, a set of domain-independent features are extracted to predict subsumption relations. The features are founded on three assumptions (topic similarity, topic distribution, and probabilistic association) based on the understanding of subsumption relations between tags.
- A Hierarchy Generation Algorithm is then proposed on top of the supervised learning model to recursively produce a hierarchy with a predefined concept. The ontology-level evaluation shows that it is particularly useful in enriching Knowledge Bases (KBs).

- A comprehensive evaluation is conducted with the large, publicly available, academic social tagging dataset Bibsonomy and three data-driven or human-engineered KBs, DBpedia, Microsoft Concept Graph (MCG), and the ACM Computing Classification System (CCS); and with three evaluation strategies, namely, the relation-level evaluation, ontology-level evaluation, and Knowledge Base Enrichment based evaluation. To our best knowledge, this is one of the largest and most systematic evaluation studies for relation learning from academic social data (*cf.* [165]); this is also the first study that evaluates the learned knowledge from tags on enriching large-scale KBs. The proposed method outperforms the state of the art in terms of F_1 score and taxonomic similarity measures when evaluated against gold standard KBs. The result is further validated through manual evaluation for Knowledge Base Enrichment.

In terms of leveraging structured knowledge,

- Two semantic-based loss regularisers are proposed to enforce the output of neural network models to conform to label similarity and subsumption relations. The semantic-based loss regularisers are independent of and can be applied to various deep learning models. Dynamic updating of the label semantic matrices adds further constraints to the semantic-based loss regularisers and allows more compatible label semantics to be learned during training.

- A Joint Multi-label Attention Network (JMAN) is proposed to model users' reading and annotation behaviour through a sentence-level, title-guided attention mechanism in the encoder. The guided attention is distinct from the original attention mechanism in the Hierarchical Attention Network (HAN) [200], through explicit modeling of the guiding source as the title instead of implicit learnable weights. This also provides insights on sentence-level attention mechanisms, which were less explored compared to word-level attention mechanisms to model social texts in recent studies [111, 192]. The title-guided attention mechanism provides a further "view" to the original attention mechanism, demonstrated through visualisation and empirical analysis.

- Extensive experiments on four datasets from real-world applications have been carried out to validate this approach. Experiments show a significant improvement of the JMAN model on the evaluation metrics with a substantial reduction of training time. A consistent performance gain was also observed when applying the semantic-based loss regularisers on Bi-GRU (Bidirectional Gated Recurrent Unit), HAN and the proposed JMAN model. Dynamic updating of the label semantics in the semantic-based loss regularisers further improved the annotation performance. The models are then analysed through the convergence analysis, parameter sensitivity analysis, multi-source components, and attention visualisation.

The above contributions in this thesis have been published or are currently under review in peer-reviewed conferences and journals, as listed in Appendix C. Implementations of the algorithms, experiments, and the datasets and results have also been openly available, as listed in Appendix B.

1.5 Overview of the Study

The overall picture of this study is illustrated in Figure 1.2. The whole thesis is centred around structured knowledge and contains two parts, learning structured knowledge and leveraging structured knowledge. Another line, driving the motivations in this thesis, focuses on tackling the unstructured characteristics of social tagging data, including noisiness, flatness, sparsity, and incompleteness, as already described in Section 1.1.

Chapter 2 introduces the concept of *structured knowledge* (illustrated in the centre of Figure 1.2), its originality, related concepts, and its different formalities from low semantics to high semantics. Folksonomies, or social tagging data, are considered as a potential source of structured knowledge. Then the chapter reviews the methods and techniques to learn structured knowledge from social tagging data, followed by leveraging structured knowledge in semantic-based, machine learning applications, especially, automated social annotation. The material in this chapter forms the basis for the studies in Chapters 3 and 4.

Chapter 3 proposes a machine learning system to learn structured knowledge from social tagging data (left side of Figure 1.2) for Knowledge Base Enrichment. The machine learning system contains five modules, Data Cleaning, Data Representation, Feature Generation, Classification and Testing, and Knowledge Enrichment. The core part of the system utilises a binary classification approach with features generated from probabilistic topic modelling. External KBs (on the bottom of Figure 1.2 are utilised for semantic grounding and instance labelling, i.e. to create labelled data for supervised learning. The key issues of noisy, ambiguity, flatness and sparsity of social tagging data are mitigated in the system. Finally, concept hierarchies and subsumption relations are created to enrich the external KBs.

Chapter 4 proposes a knowledge-enhanced and attention-based deep learning model to annotate socially shared documents with tags (see the right side of Figure 1.2). The structured knowledge of both tag similarity and subsumption relations are leveraged to improve the performance. This is mainly realised using two semantic-based loss regularisers that constrain the network output with the structured knowledge of the tags (or labels in this multi-label classification formulation). The users' reading and annotation behaviour is further modelled with guided attention mechanisms. Altogether, this chapter proposes a Joint Multi-label Attention Network for automated social annotation.

Chapter 5 concludes the thesis by addressing the research hypothesis and questions. Future studies are also discussed, mainly on the efficient learning and leveraging structured knowledge from social media data and other forms of user-generated data.

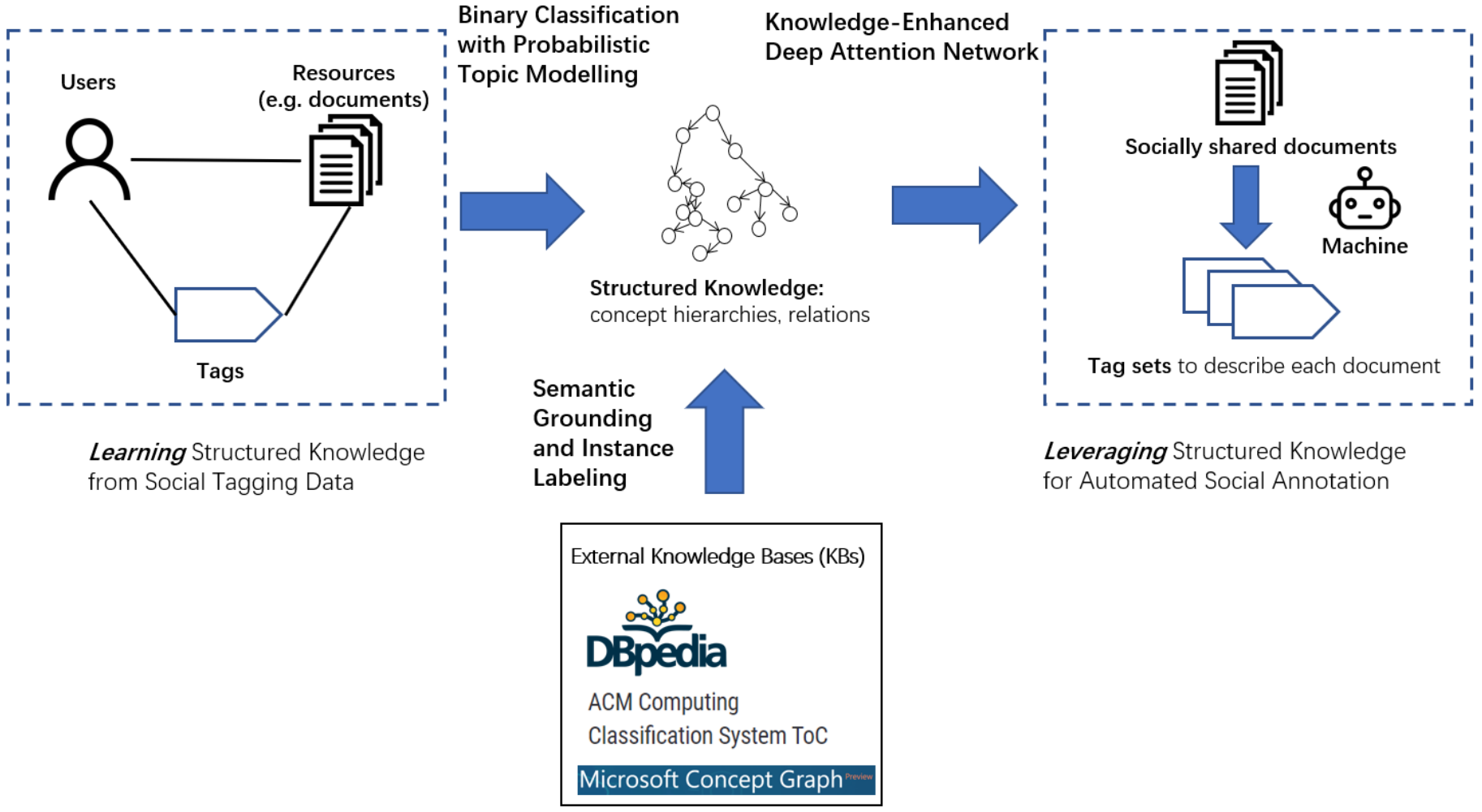


FIGURE 1.2: An overall, knowledge-centred view of research in this thesis, including *learning* and *leveraging* structured knowledge from user-generated social media data*

* The icons of "Users" and "Socially shared documents" are from Freepik in Flaticon, licensed by Creative Commons BY 3.0. `https://www.flaticon.com/free-icon/user_747376` and `https://www.flaticon.com/free-icon/documents-symbol_35920`. The icon of "Machine" is from `https://pngtree.com/free-icon/robot_378421`. The icons of the external KBs, including DBpedia, CCS and MCG, are from their official websites.

Chapter 2

Structured Knowledge: Introduction and Related Work

[Concept] structures are normally interrelated in patterns and paths so complex and so enormous no one person can understand more than a small part of them in his lifetime. – Robert Pirsig [137, p. 87]

Every ontology is a treaty - a social agreement - among people with some common motive in sharing. – Tom Gruber [74]

Structured knowledge, originally created by domain experts and potentially acquired through massive user-generated data from social media platforms, is crucial to many semantic-based applications. This chapter starts off by reviewing the relevant concepts of structured knowledge, its different types or formalities in Section 2.1. The formalities of structured knowledge are illustrated as a spectrum ranging from low semantics to high semantics. The spectrum carries the notion that Folksonomies, i.e. social tags, are most unstructured, but can be used as a source to learn structured knowledge. Based on this idea, we present the studies on *learning* structured knowledge from social tagging data in Section 2.3, including the heuristics-based, semantic grounding to external resources, unsupervised, and supervised approaches. Then we identified the studies and the research gap on enriching KBs through social tagging data. After the learning of structured knowledge, Section 2.4 provides a review on *leveraging* structured knowledge for automated social annotation as a semantic-based application. The role of structured knowledge in machine learning and semantic-based applications, especially regarding the task of automated social annotation, were discussed. In traditional approaches, the basic, tag co-occurrence relations were mostly leveraged; while for deep learning based multi-label classification, the structured knowledge is pertinent to the issue of label correlation. Section 2.5 summarises this chapter and with a discussion on the issues of current approaches for learning and leveraging structured knowledge.

2.1 Definitions and Types of Structured Knowledge

The idea of *structured knowledge* has its roots in AI, information science and semantic-based applications. There are several related concepts to structured knowledge in the thesis, Knowledge Bases (KBs), Knowledge Organisation Systems (KOSs), (web) ontologies, and Knowledge Graph (KG). From the perspective of knowledge engineering in AI, a Knowledge Base (KB) is a set of assertion about the world, which is the central component for a knowledge-based agent [148, p. 235]; the task of ontological engineering aims at representing everything in the world to facilitate knowledge-based reasoning [148, p. 437]. In the field of Library and Information Science, the concept of Knowledge Organisation Systems (KOSs) is used as a general term for all types of schemes for organizing information and managing knowledge [84, 205].

Ontologies are defined as a formal explicit specification of a shared conceptualization of a domain of interest [26, 72, 73, 166]. In the artificial intelligence and semantic Web community, ontologies vary from unstructured to structured types and form a spectrum from weak semantics to strong semantics of different formality, elaborated and illustrated in numerous occasions [15, 41, 108, 122, 156]. In the Semantic Web research, ontologies have received a lot of attention, as an enabling technology that acts as the backbone of the semantic Web, for example, ontology representation using the Simple Knowledge Organization System (SKOS). Many of the recent studies in the Semantic Web domain use ontologies of different formality (lightweight and heavyweight) to express different types of KOSs that facilitate knowledge representation and automated reasoning [15, 72, 156]. From the perspective of KOSs, ontology is regarded as the most recent or "newest" type of Knowledge Organisation Systems [84].

Another related concept is Knowledge Graph (KG), which is generally equivalent to large scale KBs, but focusing on the notion of "graphs", which are multi-relational, formed by entities and relations commonly represented as RDF (Resource Description Framework) triples [189]. KG, in its original, narrower sense, is a KB used by Google and its services to facilitate search discovery with knowledge from users' collective intelligence from the Web [155], which relies both on crowdsourced information and on automatic extraction of entities and facts from the Web. The project related to the automatic extraction of entities and facts is Knowledge Vault [50]. The idea of KG has been gradually developed to a more general term to represent large scale KB to support many semantic-based applications [189].

In this thesis, we use the term, structured knowledge, as a more abstract term to encompass the different concepts above. This term highlights the idea of "structured" being opposite to the "unstructured" characteristics of the real-world user-generated textual data and human language. We introduce different types or formalities of structured knowledge, including term lists, semantic relations, concept hierarchies, taxonomies, and ontologies (in their most formal sense), and consider folksonomies as a potential source of structured knowledge. Just as ontology and KOSs, structured knowledge can be organised from low semantic to high semantics in a spectrum, as in Figure 2.1.

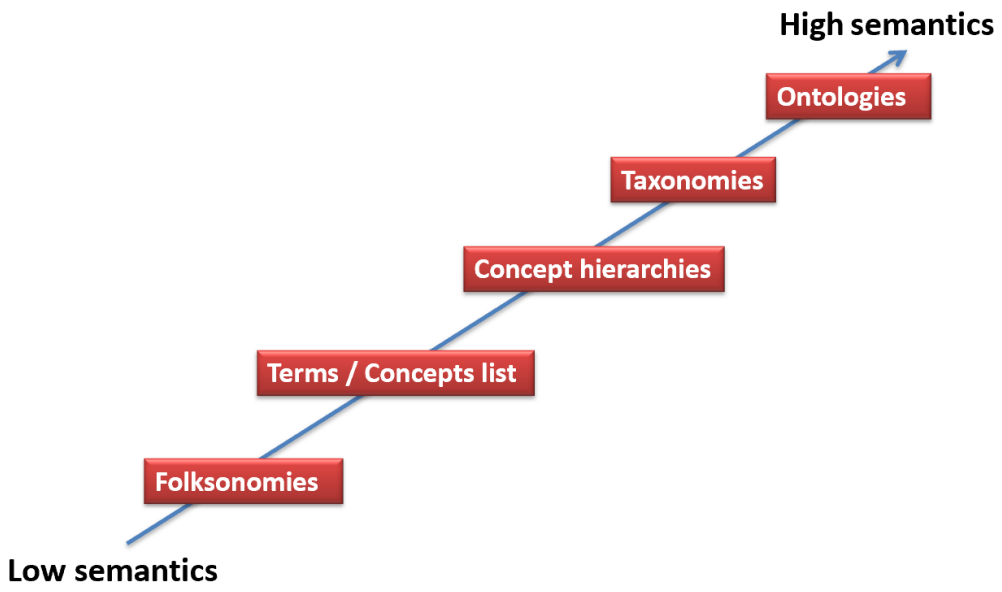


FIGURE 2.1: A simplified spectrum of (potential) structured knowledge (containing folksonomies), adapted and re-illustrated from [194, p. 319] and [108, 159, 205]

The relevant concepts of structured knowledge, covering its various types or formalities, are included in Table 2.1. The structured knowledge varies from more formal ontologies, knowledge bases, knowledge graphs, with higher semantics, to concept hierarchies, taxonomies, and various types of relations (subsumption, equivalence, association, co-occurrence), and term lists, which lower semantics. The idea of folksonomies is also included in the table, as a potential source of structured knowledge.

2.1.1 Ontologies

In its philosophical sense, ontology refers to a discipline that deals with the nature and structure of reality. In the context of knowledge representation in computer science, ontology is defined as "a formal explicit specification of a shared conceptualization of a domain of interest" in [72], originated from [73], [26] (focusing on the idea of a "shared" conceptualisation) and [166] (merging the definition of [73] and [26]). This definition captures the characteristics of "formality, explicitness, consensus, conceptuality and domain specificity" for knowledge specification [72]. According to [72], the five essential elements in a formal ontology are interrelation (relation between concepts), instantiation (assigning individual objects to classes), subsumption (is-a relationship), exclusion (is-different-from relationship) and axiomatization (complex statement about a domain). In the spectrum of formality of ontologies, similar to the spectrum in Figure 2.1, lightweight ontologies are less formal and process no or few axioms [72], but are more flexible and easier to maintain and use compare to heavyweight ontologies [180]. We denote ontology as its highest formality in this thesis.

Table 2.1: Concepts related to structured knowledge

Concepts	Definitions	Citation
Knowledge Base (KB)	A set of formal sentences or assertion about the world, which is the central component for a knowledge-based agent.	[148, p. 235]
Knowledge Organisation Systems (KOSs)	A general term to encompass all types of schemes for organising information and managing knowledge.	[84, 205]
Knowledge Graph	Large scale KBs, focusing on the notion of "graph", which are multi-relational, formed by entities and relations.	[50, 189]
Ontology	A formal explicit specification of a shared conceptualization of a domain of interest. Ontologies can have different formalities (from lightweight to heavyweight).	[26, 72, 73, 166]
Taxonomy	The type of controlled vocabulary where all the terms are connected by means of any structural model (hierarchical, tree, faceted,...) and specially oriented to browsing, organisation systems and search of contents of the web sites.	[29]
Concept Hierarchy	A set of concepts that are organised in a hierarchical fashion, typically with a subsumption relation. They are used as the backbone of ontologies.	[72]
Subsumption Relations	Also called taxonomic, "is-a", or hypernym-hyponym relations, expressing the abstraction of concepts. Subsumption relations are a type of paradigmatic relation and can be defined logically (extensionally and intensionally), collocationally, and componentially.	[40, 72, 94, 164]
Equivalence and Association Relations	Relation of words denoting the same concept (equivalence), similar or related concepts (association). These aspects of word meaning can be computationally captured through *word similarity* and *word relatedness* measures.	[27, 94, 164]
Co-occurrence Relations	Relation of words occurring nearby in a document, forming typical syntagmatic relations (compared to paradigmatic relations). Tag co-occurrence is a common relation in folksonomies.	[164]
Term List	Accepted terms with clear definitions of their senses.	[84]
Folksonomy	The result of personal free tagging of information and objects (anything with a URL) for one's own retrieval, which provides empirical material to elicit semantics and to learn structured knowledge.	[58, 124, 135, 164, 165, 184]

2.1.2 Concept Hierarchies and Taxonomies

Concept hierarchies represent a set of concepts that are organised in a hierarchical fashion, typically with a subsumption relation. They are frequently used as the backbone of ontologies [72]. Taxonomy is the type of "controlled vocabulary where all the terms are connected by means of any structural model (hierarchical, tree, faceted,...) and specially oriented to browsing, organisation systems and search of contents of the web sites" [29]. Taxonomies are well-structured, hierarchical and sometimes exclusive schemes to organise knowledge, such as the Dewey Decimal classification, and the personal or organizational file systems [66]. They can be transformed or "re-engineered" into formal lightweight, terminological ontologies [63, 182]. Building a taxonomy is resource-demanding and often needs constant manual maintenance, for example in the organisational context [36].

2.1.3 Subsumption Relations

Both concept hierarchies and taxonomies are essentially formed by subsumption relations. Subsumption relations, or is-a, hypernym-hyponym relations, express the abstraction of concepts [72, 164]. Subsumption relations are a specific type of hierarchical relation, which also includes part-of relations (or meronym-holonym) relations and instance relations [164]. A subsumption relation is a paradigmatic relation, meaning that the two words should fit into the same grammatical slot or be of the same semantic type, different from syntagmatic relations thar exist between concepts in specific documents or other contexts like windows (such as *co-occurrence relations*).

Subsumption relations can be defined logically (extensionally and intensionally), collocationally and conponentially [40]. In his first definition, hyponymy was conceptualised in a **logical** way both extensionally and intensionally. If X is a hyponym of Y, extensionally, iff $\forall x[X'(x) \rightarrow Y'(x)]$, but none of the form $\forall x[Y'(x) \rightarrow X'(x)]$, where X' and Y' are the logical constants corresponding to the concepts X and Y, and x can be understood as an instance object; intensionally, iff F(X) entails, but is not entailed by F(Y), where F(-) is a sentential function satisfied by X or Y. The second approach to define hyponyms utilises the **collocational** property. The more collocated a word, the more restricted it is through the collocational normality, then the more specific it is. "X is a hyponym of Y iff the normal context of X is a subset of the normal context of Y". The third idea for defining hyponymy is **componential**. "X is a hyponym of Y iff the features defining Y are a proper subset of features defining X". Based on this idea, analysing the degree inclusion between X and Y, "prototypes" of characterisation of hyponymy were proposed [40, 76]. Stock [164] pointed out a case that holds in most cases: there is reciprocity between the extension and intension of concepts in a hierarchical chain, in other words, specific terms, with further restrictive properties, tend to have less number of objects than general terms. The study in [49] associates the above

definitions of subsumption relations to computational rules to derive subsumption relations; meaningful hierarchies can be generated, however, further improvement could be achieved through combining the rules or features to learn subsumption relations.

2.1.4 Equivalence and Association Relations

Besides subsumption relations, the other two types of paradigmatic relations are equivalence and association relations [164]. Equivalence relations between words denote the same concept, for example, different variants of words, *autumn* and *fall*, and abbreviations, *information retrieval* and *IR*. Association relations denotes a wealth of specific semantic relations between words that are not simply equivalence or subsumption relations [164], for example, "is useful for", "located in", "derives from", etc.

The equivalence and association relations can be loosely captured through *word similarity* and *word relatedness* in natural language processing [94]. Word similarity measures how *similar or equivalent* two words are, or to what degree the words can substitute each other in context [94]. Word relatedness characterises a larger set of potential relations and defines how *associated* two words are [94]. The similarity and relatedness between words can be computed based on vector semantics and thesaurus such as WordNet[1] [27, 94].

2.1.5 Term Lists

Term lists, such as for example, glossaries and gazetteers, are distinct from folksonomies, and they include widely accepted terms with clear definitions of their senses [84], rather than undefined terms with ambiguous meanings. Therefore, term lists are considered more structured than folksonomies, although less structured than concept hierarchies and taxonomies.

2.2 Folksonomies: a Potential Source of Structured Knowledge

Folksonomies are defined as the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval [184]. Formally, folksonomies can be described as a collection of tuples, $\mathbb{F} := \langle U, T, R, Y \rangle$, where U, T and R are finite sets representing *users*, *tags* and *resources*, respectively; Y is a ternary relation among them, $Y \subseteq U \times T \times R$ [154].

While some studies also categorise folksonomies as a type of structured knowledge, in fact, they are highly unstructured and uncontrolled, as users can freely add tags to annotate resources without constraints in most social tagging systems [6, 184]. Therefore, folksonomies inherit many of the problems in human natural language. Compared to ontologies, folksonomies lack a uniform representation to facilitate their sharing and

[1] `https://wordnet.princeton.edu`

reuse [58]. Without special processing and treatment, social tagging systems are not able to discriminate different noisiness, for example, semantic and syntactic variations of tags.

The advantage of folksonomies is that they are contributed by communities, representing a social dimension with a free form tagging functionality, and thus contain many emerging terms or neologisms that can potentially complement the controlled vocabularies [105, 116]. Folksonomies can support resource classification [215] and recommendation [22]. Folksonomies provide rich tag co-occurrence relations, which are empirical sources to exploit paradigmatic relations such as subsumption, equivalence and association relations from tags, as elaborated in [133, p. 222-p. 224] and [164]. Studies have thus identified folksonomies as a source for eliciting semantic relations and developing structured knowledge [58, 165], or even as a lightweight ontology [124]. The rich metadata in folksonomies, however, are accompanied by some inherent issues as other user-generated social media data.

2.2.1 Unstructured Characteristics of Folksonomies

From the spectrum of (potential) structured knowledge in Figure 2.1, we can observe that folksonomies are the most unstructured type of sources, having the lowest semantics, compared to the presented structured knowledge.

As described in Section 1.1 and reviewed in the literature [6, 90], the unstructured characteristics of folksonomies includes noisiness, flatness, sparsity, and incompleteness.

Noisiness The noisiness of tagging data is an inherent characteristic of human language [135, p.218-228]. Without the controlling of vocabularies and the definitions of meanings, tags have various morphological forms and ambiguous meanings. Multiple words in a tag ("speechanalysis", "Time-series_analysis"), tags with special characters ("Autoantibodies/*analysis/drug"), polysemous tags ("apple"), multilingual tags ("Datenanalyse"), variations and misspelled tags ("analysis", "analyses", "analysed", "analys"), nonsense tags ("28A75"), and other forms of noisy tags can occur, as summarised in [90] and [6].

Flatness The unstructured characteristics of folksonomies are also related to their flat form. No explicit relations among tags are defined in folksonomies, different from more structured forms of knowledge such as concept hierarchies and ontologies [134, p. 222-223]. This prevents efficient usage of tags to support navigation, browsing, retrieval and recommendation of resources.

Sparsity Due to a large number of tags, users and resources in social tagging data, the interaction among them is sparse. This is typically for academic social tagging data (annotating academic resources, such as Bibsonomy, CiteULike, and Connotea), which has more unique tags and slow accumulation of tags than general social tagging data (annotating general websites, such as Delicious) [51]; the *agreement* between users is also less in academic social tagging data, which is a critical

shared context to induce relations among tags [80]. The high sparsity of tagging data is also related to the lack of contextual or sentential information in tag sets, which is very different from any other types of user-generated texts in social media platforms, such as (micro-)blogs and online comments [192].

Incompleteness The incompleteness issue is related to sparsity, which highlights the phenomenon that many resources are not associated with tags. For example, in the social question and answering (Q&A) website Zhihu, more than 18% of questions are not associated with any tags, as reported in [130]. In the micro-blogging website Twitter, only 10% to 15% of tweets are associated with at least one hashtag [97, 191]. The incompleteness issue of tagging data may be due to the fact that annotating content requires cognitive effort and can be time-consuming.

On the other hand, folksonomies are valuable metadata collaboratively created from users, thus are highly potential to enrich the existing structured knowledge. This motivates a research direction to learn structured knowledge from folksonomies, i.e. social tagging data.

2.3 Learning Structured Knowledge from Social Tagging Data

Deriving structured knowledge from a data-driven perspective has been a vivid research area named *ontology learning*. Ontology learning is a necessary part of ontological engineering (or ontology engineering) that represents entities and relations from the real world, as massive data are available and learning from them can indeed reduce the cost to create and maintain structured knowledge [37]. While some research focused on ontology engineering from source with higher semantics, including taxonomies, thesaurus (i.e. controlled term list with subsumption, equivalent and associative relations) and lexica (i.e. hierarchically organised controlled vocabularies with meaning and linguistic behavioural information, such as WordNet) [182], a substantial part of studies explored learning structured knowledge from social tagging data, or folksonomies.

Learning structured knowledge from folksonomies suffers from the unstructured characteristics associated with social tagging data, as described above. The study [48] categorises research in this area to learn term lists (or concept lists) and to learn relations from tags. To learn term lists or concept lists, studies applied some unsupervised [6] and supervised approaches [92] leveraging external KBs or heuristics, see the review in [48]. We will focus on presenting the studies regarding the latter, learning relations from tags, which is inherently more complex and share a similar category of methods as learning term lists from tags. Existing methods for learning structured knowledge from social tagging data can be broadly categorised into four classes: heuristic-based, semantic grounding to external resources, unsupervised learning and supervised learning.

2.3.1 Heuristics-Based Methods

Heuristics-based methods mostly make use of heuristics to infer relations with respect to pre-defined rules. A common heuristic is the generality measure based on set inclusion. The work in [124] detected subsumption relations between tags using the inclusion of user sets, within a dataset crawled from the general domain social tagging system Delicious[2]. The study in [123] further defined a metric called *inclusion degree* and *generalisation degree* and automatically generates hierarchies using graph-pruning algorithms. Graph centrality is another well-known heuristic in the literature [14, 80]. The research in [80] induced a taxonomy using a greedy search algorithm with the degree centrality of tag nodes in a tag similarity graph. The study in [14] extended this approach with sense disambiguation and applied betweenness centrality on a tag-tag co-occurrence network. The work in [165] evaluated both methods proposed in [14, 80] and validated the usefulness of graph centrality in creating taxonomies from tags. Later research extended the graph centrality by either combining semantic grounding to external lexical resource to increase accuracy [3] or using new centrality measures on a weighted tag-tag graph [32]. This class of method heavily relies on co-occurrence information and may not derive accurate subsumption relations [58]. The co-occurrence-based heuristics are sensitive to data sparsity; with the graph-centrality measure, it is more difficult to generate a hierarchy from the academic social tagging data such as CiteULike[3] than from the general domains like Delicious, as the data in the former has lower *agreement* among users, and lower data *density* and *overlap* [80]. This problem has also been statistically analysed in [51] and discussed in [183]. Thus for the sparse social tagging data, especially in the academic domain, such as Bibsonomy, CiteULike and other narrow folksonomies, the co-occurrence-based heuristics are unsuitable.

2.3.2 Semantic Grounding to External Resources

Methods of semantic grounding to external resources based methods attempt to match tags to entities in external KBs in order to find semantic relations. The work in [47] mapped social tags to concepts in WordNet to extract relations. However, WordNet is a relatively static resource and only less than half (48.7%) of the tags could be directly matched according to the study in [7]. The work in [59] used DBpedia and its inter-connected datasets in the Linked Open Data Cloud[4] to ground tags and populate an ontology. In general, it is however difficult to choose the one with the right sense due to the lack of tagging context. This is because users' collective tagging process is very different from that of lexicographers or domain experts. This tag sense disambiguation problem has been discussed in [7, 47, 59]. Even if a tag can be lexically matched to a concept in external resources, it is uncertain that its intended meanings coincide with each other [33]. A potential solution for tag sense disambiguation is to use intelligent

[2] `https://del.icio.us`
[3] `http://www.citeulike.org/`
[4] `https://lod-cloud.net`

tools and contextual sources for semantic grounding, for example, the work in [4] utilises Google search[5] and Wikipedia articles[6] to disambiguate tag senses and establish tag-tag relations.

2.3.3 Unsupervised Learning

Unsupervised learning based methods mostly use various clustering or dimensionality reduction techniques. The research in [211] proposed a hierarchical clustering model based on *Deterministic Annealing* to generate subsumption structures from social tagging data in Delicious and Flickr. However, the model could not clearly discriminate subsumption, related and equivalent relations. Another clustering based method using k-means [165] showed that it did not perform better than the graph-based methods [14, 80]. Other unsupervised methods attempt to find low dimensional representations of data items to discover semantic patterns. *Probabilistic topic models* [18, 19], such as *Latent Dirichlet Allocation* (LDA), are a type of generative model used to discover themes from a large collection of documents. The study reported in [103] proposed a hybrid approach utilising graph-based heuristics with contextual information inferred, using LDA, from a web corpus to learn domain ontologies from tags. The study in [190] applied LDA to a collection of abstracts of scientific publications and represented concepts through a "fold-in" process. It proposed a metric, *Information Theory Principle for Concept Relationship*, to determine subsumption relations based on the asymmetric difference of the *Kullback-Leibler Divergence* of topic distributions. The work in [174] also defined similar metrics using a *Tag-Topic model*. A common issue of these methods is whether using the divergence measure is precise enough to determine relations for tagging data.

2.3.4 Supervised Learning

Supervised learning methods have also been proposed. The study reported in [212] used a binary classifier to generate a taxonomy from Stack Overflow tags. Both co-occurrence-based features and topic-based features were considered. The feature generation process has leveraged the textual information of resources (such as questions and wiki descriptions of tags on Stack Overflow), which may be unavailable in other types of social tagging data. Moreover, the proposed topic-based features may not be fine-grained enough to represent the topic information in social tags to quantify the subsumption relations. Work reported in [144] combined several popular co-occurrence-based feature extraction mechanisms to develop a binary classifier. The mechanisms considered included support and confidence [150], cosine similarity, set inclusion and generalisation degree [123], mutual overlapping [28] and graph-based taxonomy search adapted from [80]. It is reported that combining these heuristics in a classifier significantly increased the F_1 score in relation-level evaluation. However, the method has the same drawbacks

[5] `https://www.google.com/`
[6] `https://www.wikipedia.org/`

as other co-occurrence-based methods in that it does not take into consideration the complex meanings of tags and suffers from the data sparsity problem.

2.3.5 Knowledge Base Enrichment from Folksonomies

One key aspect of learning structured knowledge from tags is to elicit new semantics to enhance existing structured knowledge, such as KBs or ontologies like DBpedia and thesaurus like Medical Subject Headings[7]. While many studies have used KBs or ontologies to enrich folksonomies [8, 57, 59], less research has explored the opposite case: using folksonomies to enrich KBs. However, it is generally agreed that folksonomies represent users' terminologies and can be extracted to enrich KBs. This has been validated through comparison studies between folksonomies and controlled vocabularies. The work in [105] compared the academic tags in CiteULike with Medical Subject Headings and shows they have a highly distinct lexicon and viewpoints. The study from [116] compared the Librarything[8] tags with the Library of Congress Subject Headings[9] and shows little overlap between ordinary users' and experts' vocabularies.

The work in [5] proposed the idea of "Folksonomised Ontology", which is a fused terminological ontology based on folksonomies and existing KBs. It suggests the so-called "3E" techniques (Extraction, Enrichment, Evolution): (1) preprocessing the social tagging data to obtain a cleaned tag set (Extraction); (2) matching the tag concepts to KBs and enrich the relations in KB with co-occurrence weights (Enrichment); and (3) using tag-tag subgraphs (or "tagsets") to enrich relations in existing KBs (Evolution). Co-occurrence information was primarily used to discover the relations between tags. The enrichment and evolution processes in [5] require much human intervention with visualisation techniques. Similarly, another work on Knowledge Base Enrichment from tags in an e-learning environment [60] focused on designing a visual interface for educators to view and manually edit learning ontologies and used a similarity metric to suggest new concepts as learners' tags to be enriched.

2.4 Leveraging Structured Knowledge for Automated Social Annotation

Structured knowledge provides contextual background for machine learning and semantic-based applications (or applications that require knowledge or semantic information) [21]. In this section, we briefly review the semantic-based applications and some representative work on leveraging structured knowledge in these applications. This section focuses on *automated social annotation* as a representative semantic-based application that requires knowledge from social tags. Studies have leveraged co-occurrence relations of labels as knowledge for automated social annotation, more recent studies apply deep

[7] `https://www.nlm.nih.gov/mesh/meshhome.html`
[8] `https://www.librarything.com/`
[9] `http://www.loc.gov/aba/cataloging/subject/`

learning approaches which commonly formulate the task as a *multi-label classification* problem. We will first present the task, automated social annotation, especially regarding the perspective of leveraging structured knowledge in traditional and deep learning methods. Then we introduce the role of structured knowledge in multi-label classification to tackle the issue of label correlation.

2.4.1 Automated Social Annotation as a Semantic-Based Application

The study [21] is one early review summarising studies on leveraging structured knowledge to support machine learning and semantic-based applications such as text mining (including text clustering, classification, and visualisation), ontology-based similarity measuring, information retrieval, etc. Most approaches in [21] aim at enhancing the traditional bag-of-words representation with ontology-based representations. As this type of approach relies on matching concepts to enhance word representation, the results are largely affected by the accuracy of concept matching and the disambiguation [86]. An early integrated and knowledge-centred framework for text classification was proposed in [20]. The study applies several unsupervised and supervised methods to learn structured knowledge from texts, and then, utilises the learned knowledge to support text categorisation through enhancing the bag-of-words representation [20]. The review of [61] summarised the studies on semantic-based recommender systems that leverage social tags to enhance the performance; most studies simply applied the tag set as a list of flat terms, or perform clustering on the tags without explicitly inferring the relation among them, to represent items for representation. Structured knowledge of higher semantics, such as concept hierarchies and ontologies were not much leveraged in studies review in [61]. A recent trend is to utilise large scale KBs or KGs with end-to-end deep learning approaches to support semantic-based applications, such as item recommendation [188, 206], sentiment analysis [101] and task-oriented dialog systems [118].

Among many semantic-based applications, *automated social annotation* can support users' tagging process, reduce their cognitive overhead, address the incompleteness issue of tagging data and help produce more stable, higher quality folksonomies in social media platforms [11, 90, 130]. While tags are originally created by users, it is natural to consider, with a collection of documents and their associated tags, whether it is possible to automatically annotate new documents and the previously nontagged documents. This task becomes especially important when a substantial amount of socially shared documents online are not annotated with any (hash-)tags, e.g. over 18% of the questions in Zhihu [130] and at least 85% of the microblogs (or tweets) on Twitter [97, 191]. The task is closely related to *tag recommendation*, which aims at suggesting tags for existing or previously unseen resources to facilitate users' tagging [11]. The study in [11] classifies tag recommendation as either *object-centred* or *personalised*. Object-centred recommendation predicts a set of tags that are related to or can describe an object regardless of the target user. This type of recommendation aims at enhancing the quality

of tagging and thus can benefit information retrieval in general. Another type of recommendation, personalised recommendation takes the users' interest into consideration. Automated social annotation can thus be considered as an object-centred type of tag recommendation. The sections below review the methods and techniques for automated social annotation and tag recommendation, with a focus on how structured knowledge was leveraged in these applications.

2.4.2 Knowledge as Tag Co-occurrence Relations

In social tagging systems, various methods and techniques have been proposed for tag recommendation, as reviewed in [11], including *tag co-occurrence-based*, *content-based*, *matrix factorisation based*, *clustering-based*, *graph-based*, *learning to rank based* approaches. In social Q&A sites, existing research explores the annotation of descriptive tags for a question by the tags of its similar questions through *probabilistic hypergraph* construction, adaptive probabilistic hypergraph learning and heuristics-based tag selection [130]. In microblogging services such as Twitter, various models have been proposed for *content-based* hashtag recommendation [46, 67, 89, 110, 203, 210], that is, to suggest tags according to the textual features from the documents (or tweets). Previous research extracted *term frequency based* lexical features [203] and applied *probabilistic graphical models* [46] to suggest hashtags for tweets and recent studies are mostly focused on *deep learning* approaches [67, 89, 110, 210].

Among the studies presented above, knowledge mainly takes the form of co-occurrence relations between tags, especially regarding the co-occurrence-based approach for tag recommendation in [11], which exploits co-occurrence patterns, or association rules, of the tags of the shared resources [12, 81]. Tag co-occurrence is also used to enhance the performance of content-based approaches [12]. Tag co-occurrence patterns have also been encoded as weights between the last hidden layer and the output layer of neural networks to predict multiple labels for a document [102]. As discussed in Section 2.1.3, co-occurrence relation is a syntagmatic relation, which exists between words that appear together or nearby each other; while the subsumption relations in concept hierarchies are a type of paradigmatic relation [152, 164] which carries "tight" [164, p. 1958] and explicit semantics. Studies on automated social annotation mostly rely on the co-occurrence relations of tags, but did not leverage more explicit structured knowledge, such as similarity and subsumption relations that can be learned from social tagging data as reviewed in Section 2.3.

2.4.3 Knowledge in Deep Learning Approaches

Structured knowledge is also to be applied in deep learning approaches for automated social annotation. Recent studies adapted deep learning approaches that encode the input, with layers and nodes of non-linear activations, to a continuous representation and approximate the matching from the input representation to the label space, for microblog annotation [67, 89, 110, 210] and paper annotation [77]. Most studies with

deep learning approach formulate the automated annotation task as a *multi-label classification* problem. Under this formulation, the relations among the labels in the output space should be considered [62, 176]. This is actually the structured knowledge of the tags for automated social annotation. While many studies propose methods to incorporate knowledge in deep learning, such as continuous representations of structured knowledge (e.g. Knowledge Graph Embedding, represented by the work in [25] and reviewed in [189]) and to input structured knowledge as contextual information or as memory through attention mechanisms [24, 101], fewer studies have been on leveraging structured knowledge for deep learning based multi-label classification. This key issue of *label correlation* in multi-label classification, also pertinent to the idea of structured knowledge, is reviewed as follows.

2.4.4 Knowledge as Label Correlation in Multi-Label Classification

In multi-label classification, each instance is associated with a set of labels and the labels are correlated to each other [62, 176]. This is different from traditional single-label classification where classes (labels) are disjoint. This multi-label characteristic corresponds to the scenario of document annotation with tags, as an object is most likely annotated with several user-generated tags instead of one single tag.

Structured knowledge of the labels, i.e. their relationships or label correlation, can be exploited to improve the performance of multi-label classification algorithm [62]. In real-world data, normally with a large label size, the correlation among labels is common. In social tagging, users tend to collectively annotate tags with various semantic forms and granularities [80, 133]. According to the Bibsonomy data, many documents tagged with *machine_learning* are also tagged with *text_mining*, *svm* or *optimization*[10], which are either the related terms (*text_mining* being a related application domain), or more specifically the narrower terms (the specific algorithm *svm* and the sub-domain *optimization*).

A traditional approach for multi-label classification is to construct many binary classifiers, one for each label. This approach, called *binary relevance* or *one-vs-rest*, however completely ignores the correlations among labels [128, 209]. One main strategy to address this issue was to re-generate a feature space incorporating information on label correlation. An example was adapting discriminative classifier like Support Vector Machine (SVM) [65]. The Classifier Chain method extends this idea above one-vs-rest through incorporating the binary classification results of previous labels in a chain as features to predict the next label [141]; classifier chains can be randomised and embedded into an ensemble learning architecture [142] or mined using clustering and graph-based methods [31]. Instead of organising classifiers as a chain, the HOMER (Hierarchy Of Multilabel classifiER) approach [177] creates a tree of classifiers, based on the hierarchical structure of labels pre-learned in an unsupervised manner. *Probabilistic graphical*

[10] `https://www.bibsonomy.org/search/machine_learning`

models were also used to encode the correlation among labels, including *Gibbs Random Fields* [138] and *Bayesian Networks* [207].

For *deep learning* approaches, which report superior performance over other methods [208, 209], there are still inadequate studies considering the issue of label correlation. Neural network models adapted for multi-label classification usually represent each label with a one-hot representation, as an orthogonal vector in the label space, and each label set with a *multi-hot* representation, e.g. [0 1 0 1 1] in a 5-dimensional label space, as in [77, 89, 110, 128, 210] (see also Section 4.2). This, however, assumes independence among labels, i.e. not leveraging any structured knowledge regarding the labels.

One approach to leverage structured knowledge of the labels in neural networks is through *weight initialisation* [102]: initialising higher weights for some dedicated neurons, that each represents a co-occurring pattern among labels, between the last hidden layer and the output layer. This idea is extended in [10] to include subsumption relations among labels. It is, however, difficult to interpret how the randomly chosen "dedicated" neurons really work in neural networks to leverage relations between labels. Computationally, it is also ineffective (if not infeasible) to place many neurons, equal to the large number of co-occurring patterns, in the last hidden layer for weight initialisation. Another approach is through *architecture adaptation* for hierarchical multi-label classification as in [193]. The study [193] explored tree-like architectures to organise neural networks as a chain for hierarchical label prediction: assigning a chained feed-forward neural network for each layer in a label hierarchy. Similar to the idea of assigning dedicated neurons, this cannot be easily scaled to a massive number of label similarity and subsumption relations. More generalisable methods that can leverage structured knowledge of large scale for multi-label classification are expected.

2.5 Summary and Discussion

In this chapter, we have introduced the concept, formalities and the spectrum of structured knowledge. Based on the idea that folksonomies are highly unstructured but a rich source to learn structured knowledge, we reviewed the studies and methods to learn structured knowledge from social tagging data. The current approaches can be categorised as heuristics-based, semantic grounding to external sources, unsupervised learning, and supervised learning based methods. While heuristics-based methods are efficient, they heavily rely on co-occurrence information to infer the subsumption relations, which are not explicit and accurate enough and are not suitable for sparse tagging datasets. The semantic grounding based approaches can infer explicit relations, but suffer low coverage and low accuracy when matching the tags to concepts from external KBs. The machine learning based approach so far, both unsupervised and supervised, either cannot infer clear semantic relations or the features are not fine-grained enough to infer the relations. The most challenging and unsolved issues to learn structured knowledge from tags are, therefore, the representation of the highly ambiguous meaning

of tags and the quantification of their semantic relations to yield more accurate machine learning models. Besides, few studies explored the enrichment of KBs from the learned tag structures. In the next Chapter (Chapter 3), we will present a supervised machine learning system, that articulates the meaning of tags using a set of features on top of tag representation with probabilistic topic models, to learn structured knowledge for Knowledge Base Enrichment.

Studies have also explored how to leverage structured knowledge to support and improve the performance of various semantic-based applications. One typical application is automated social annotation that can greatly address the incompleteness issue of current tagging data and can maintain tagging quality. Knowledge regarding the labels (tags) plays a key role in the task of automated social annotation. Most approaches rely on using co-occurrence relations of the tags. For the recent, deep learning approaches, research formulates the task as a multi-label classification problem. Structured knowledge in this context is pertinent to the label correlation issue in multi-label classification. However, although many studies have attempted to incorporate knowledge into deep learning in general, few have explored the use of structured knowledge to address the label correlation issue. In Chapter 4, a novel deep learning model will be introduced to leverage both similarity and subsumption relations in multi-label classification for automated social annotation.

Chapter 3

A Machine Learning System to Derive Knowledge from Tags

The acquisition of any knowledge is always of use to the intellect, because it may thus drive out useless things and retain the good. For nothing can be loved or hated unless it is first known. – Leonardo da Vinci [131, p. 293]

An experiment is never a failure solely because it fails to achieve predicted results. An experiment is a failure only when it also fails adequately to test the hypothesis in question, when the data it produces don't prove anything one way or another. – Robert Pirsig [137, p. 95]

Deriving the "collective intelligence" from user-generated social media data is a challenging and complex process, due to unstructured characteristics of data and the inherent difficulties in learning structured knowledge. In this chapter, we present a novel machine learning system to learn structured knowledge from social tags, based on assumptions founded on probabilistic topic modelling of tags. We introduce the overall system architecture in Section 3.1. The machine learning system has five modules, namely, Data Cleaning, Data Representation, Feature Generation, Classification and Testing, and Knowledge Enrichment. To deal with the noisiness and the sparsity of social tags, as the first part of the system, a data cleaning module is proposed to extract tag concepts from the raw tags, in Section 3.2. To predict accurate subsumption relations between tags, the system takes a supervised learning approach, with features regarding a pair of tag concepts and a context tag concept. To capture the ambiguity of meanings of tag concepts, the unsupervised, probabilistic topic modelling approach is adapted for data representation in Section 3.3, which also further mitigates the data sparsity issue. Then the proposed features, to quantify subsumption relations between tag concepts under a context concept, are presented in Section 3.4, based on a formal set of assumptions on deriving subsumption relations. The instance labelling, data collection and

creation process for the Classification and Testing module are presented in Section 3.6. Once the machine learning models are trained and tested, tag concept hierarchies can be formed through a Hierarchy Generation Algorithm which predicts and organises tag concepts progressively from top to down into hierarchies, in Section 3.5, as a part and a prerequisite of Knowledge Enrichment.

Evaluation is a crucial scientific process for studies in the area of machine learning, data mining, and natural language processing. The other contribution in this chapter is the formal assessment of the quality of the learned structured knowledge from user-generated data. The proposed evaluation strategies contain three parts, relation-level evaluation, ontology-level evaluation, and Knowledge Base Enrichment based evaluation. The first two evaluation strategies assess the accuracy of the learned relations and hierarchies; and the Knowledge Base Enrichment based evaluation, as a part of the Knowledge Enrichment module, focuses on a manual assessment of the new concepts and relations, not included in the existing KBs. The evaluation strategies, results, discussions, and visualisation of the learned structured knowledge, i.e. subsumption relations and concept hierarchies, are presented in Section 3.6.

Summary and further discussions of the methods and limitations are in Section 3.8.

3.1 Definition, Problem Formulation and Overview of the System

The proposed system focuses on learning relations, especially subsumption relations (introduced in Section 2.1.3 as a type of structured knowledge), from social tagging data. The task can be formulated as a supervised learning problem. Before presenting the learning framework, we first introduce some formal definitions used in this study.

We formally review the definition of folksonomies again and extend to the ideas of cleaned and structured folksonomies. Folksonomies can be described as a collection of tuples, $\mathbb{F} := \langle U, T, R, Y \rangle$, where U, T and R are finite sets representing *users*, *tags* and *resources*, respectively; Y is a ternary relation among them, $Y \subseteq U \times T \times R$ [154]. As folksonomies are noisy, they need to be cleaned and variants of tags need to be identified. A cleaned folksonomy is denoted as $\mathbb{F}^{clean} := \langle U, C, R, Y \rangle$, where the original T is transformed to a new finite set C representing *tag concepts*. Each element in C is a group of tags considered equivalent. The task is to learn subsumption relations from the cleaned folksonomies and finally transform these to structured folksonomies, $\mathbb{F}^{str} := \langle U, C, R, Y, \prec \rangle$, where $\prec$ represents the set of learned subsumption relations, $\prec \subseteq C \times C$.

As a simple example, suppose that the raw folksonomy $\mathbb{F}$ contains four tuples regarding two users (u1 and u2) and two resources (r1 and r2), $\mathbb{F}$ = {<u1,semanticweb,r1>, <u1,socialsoftware,r1>, <u2,ontologies,r2>, <u2,semantic-web,r2>}. To create $\mathbb{F}^{clean}$, the tag variants 'semanticweb' and 'semantic-web' will be unified to a standard form of 'Semantic_Web', and 'socialsoftware' to 'Social_Software' (see the examples in Figure

3.2). To form $\mathbb{F}^{str}$, the subsumption relation <ontology $\rightarrow$ Semantic_Web> should be specified.

The subsumption relation learning process can be formalised as a *binary classification* problem, which is a type of *supervised learning* in machine learning, where *labels* belong to two categories [127, p. 3-6]. Based on [127, p. 7] and adapted from [144], the problem can be defined as follows: Let $\mathcal{X}$ be the set of instances or triples $< C_a, C_b, C_r >$ in the input space and $\mathcal{Y} = \{0, 1\}$ be the set of positive and negative labels for the instances. Each instance is represented as a vector, $\vec{x_i} = (f_1(C_a, C_b, C_r), ..., f_m(C_a, C_b, C_r))$, $(C_a \neq C_b)$, where C_a and C_b are two concepts whose relation is to be determined, and C_r denotes the context of the instance. C_r can be either the direct or indirect parent concept of C_b. The identifiers f_1 to f_m represent a set of different feature extraction functions (which, in this research, are founded on assumptions based on probabilistic topic modelling). The objective is to learn a function $h : \mathcal{X} \rightarrow \mathcal{Y}$ to predict the subsumption relations between tags.

Figure 3.1 provides an overview of the proposed learning framework, which consists of five blocks:

1. Data Cleaning: transforming $\mathbb{F}$ to $\mathbb{F}^{clean}$ by unifying tag variants and removing infrequent tags;

2. Data Representation: using probabilistic topic models to represent each tag concept as a distribution of latent topics in a lower dimensional semantic space;

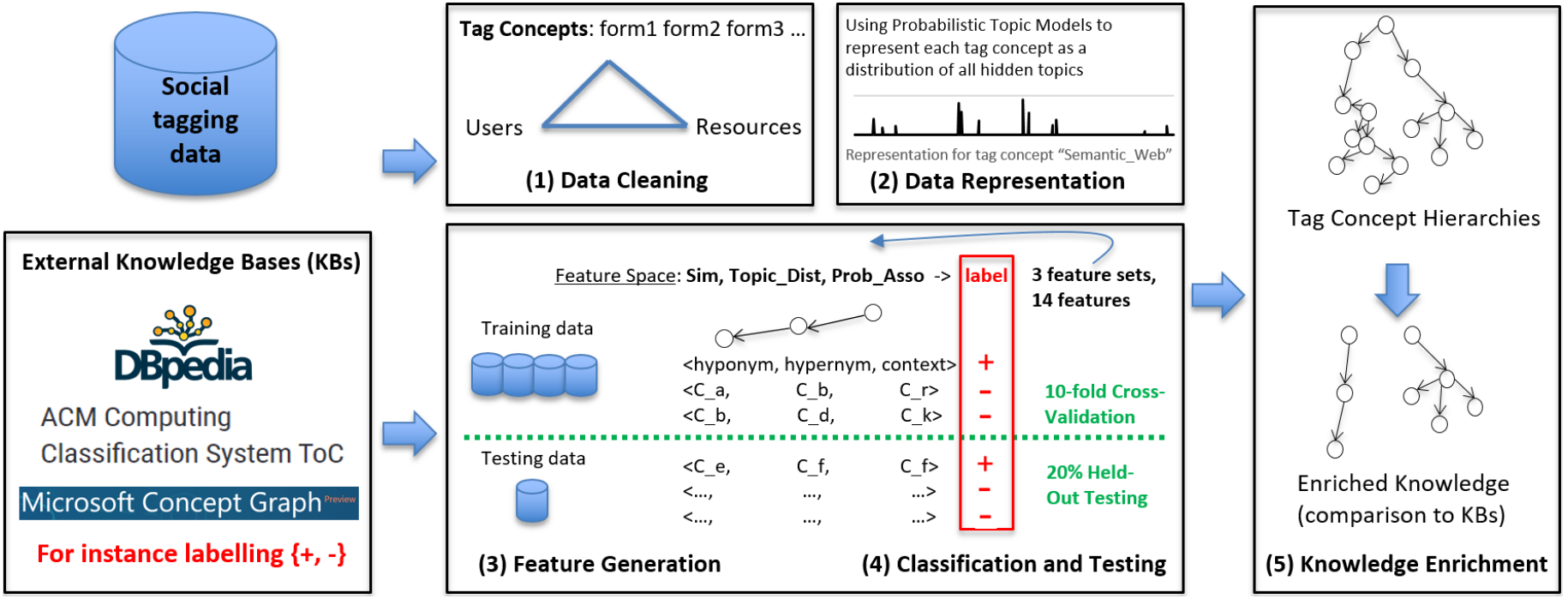
3. Feature Generation: generating features founded on assumptions of topic similarity, topic distribution and probabilistic association to measure the degree of subsumption between tag concepts given a context tag concept;

4. Classification and Testing: automatic creation of training and testing data through semantic grounding to external KBs, followed by training and testing of the classification models;

5. Knowledge Enrichment: using a Hierarchy Generation Algorithm to transform $\mathbb{F}^{clean}$ to $\mathbb{F}^{str}$, followed by discovering new concepts and relations through comparing to existing KBs. At the end, the results are presented to human domain experts for verification.

It should be noted that, the input to the Feature Generation module contains triples of tag concepts. A triple is labelled as true if given the context tag concept C_r, the tag concept C_a is a direct hyponym (narrower concept) of C_b, or in other words, C_b is a direct hypernym (broader concept) of C_a. In this case, there is an established subsumption relation in this triple; in other cases, the triple is labelled as false. The same rules apply to the prediction of subsumption relations.

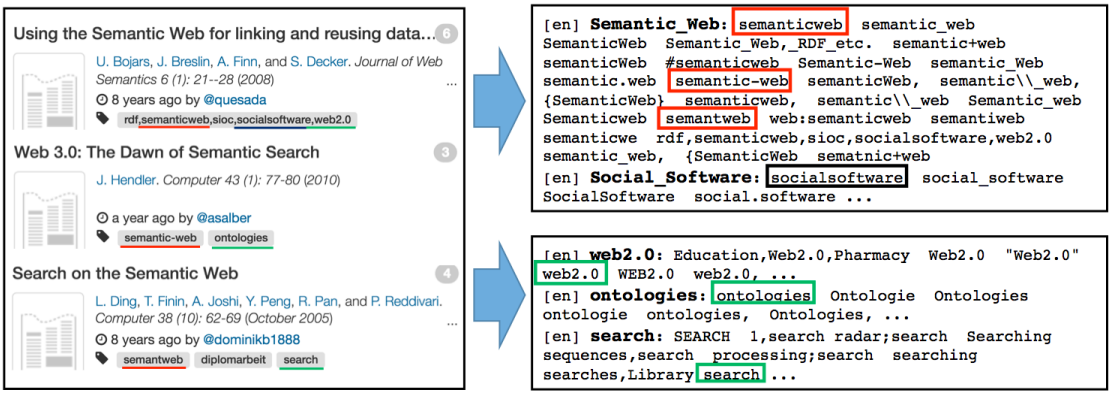
FIGURE 3.1: Architecture of the system to learn subsumption relations and concept hierarchies from social tagging data



3.2 Data Cleaning

The Data Cleaning module aims at reducing the different types of noisiness and addressing the sparsity in social tagging data (see Section 2.2.1). The module cleaned social tags using simple morphological and statistical methods in four steps: (i) specific character handling, (ii) multiword and single tag group extraction, (iii) tag selection using metrics. Specific character handling will remove nonsense characters and separate individual tags from a compounded tag. Tag selection will ensure meaningful and high quality tags being selected. We mainly introduce the steps (ii) here regarding the extraction of tag concepts and explain more details in Section 3.6.

FIGURE 3.2: Extracting tag concepts using the Data Cleaning module, from the Bibsonomy Dataset: The underlined tags with coloured lines (on the left) are grouped to form several tag concepts (on the right), either a multiword tag concept (in the upper right black box) or a single tag concept (in the lower right black box); the standard tag concepts are marked in bold font.



As shown in Figure 3.2, a tag concept in C is regarded as either a multiword tag group (on the upper right) or a single tag group (on the lower right), extracted from the raw tag sets. *Multiword expressions*, for example "Semantic_Web" and "Social_Software", are words commonly used as composition of multiple lexemes, and should be treated as a whole for computation [139, p. 31]. After the handing of special characters, a multiword tag can be recognised by whether there is an underscore (_) within the tag. Different forms (tags surrounded with coloured boxes) of a multiword tag can be grouped together (including those tags which do not have an underscore inside) according to the *minimum edit distances*, *Levenshtein distances*, between tags [93, 109]. A standard form (marked in bold) of each multiword tag group is selected based on the *user frequency* or *inter-subjectivity* [90], i.e. the tag form with highest frequency annotated by users. The remaining tags are single word tags, which are then lemmatized and grouped based on their lemma. To filter out insignificant tags, empirical results show that user-based metrics tend to produce the best tag quality selection result [153]; we set a threshold on user frequency of tag concepts to filter them. More details are in Section 3.6.

3.3 Data Representation

Social tagging data, if projected along the dimensions of the resources (R) or the tags (T), have very high dimensionality and are extremely sparse. To address the sparsity problem as well as the ambiguity of meaning in tags, it is necessary to reduce the dimensionality of the tagging data. Each resource, $r \in R$ in $\mathbb{F}^{clean}$, is initially represented as a set of "bag of tags" (here tags mean tag concepts in $\mathbb{F}^{clean}$), analogous to the *bag of words* model in Information Retrieval [120, p. 107]. We wish to infer a low dimensional topic structure from the large collection of resources and tags. With Latent Dirichlet Allocation [19], we can obtain the topic assignment for each tag in all the resources, and consequently, two probabilistic distributions: the tag-topic distributions $p(C|\mathbf{z})$ (the set of all $p(C|z)$, $z \in \mathbf{z}$) which represent each latent topic as a distribution of tag concepts; and the topic-resource distributions $p(\mathbf{z}|R)$ (the set of all $p(\mathbf{z}|R)$, $z \in \mathbf{z}$) which represent each resource as a distribution of latent topics.

More specifically, Latent Dirichlet Allocation [19] can be adapted to model the generation process of the whole bag-of-tags, as depicted as following, where the hyperparameters α and β are the concentration parameters for the two symmetric Dirichlet distributions; $p(C|z)$, $p(\mathbf{z}|R)$ can be approximated by Gibbs sampling [70, 79].

1. For each topic z, draw a multinomial distribution $p(C|z)$ from a Dirichlet prior β.
2. For each bag-of-tags used to annotate a resource R, draw a multinomial distribution over topics $p(\mathbf{z}|R)$ from a Dirichlet prior α.
3. For each tag concept in a bag-of-tags used to annotate a resource, there are two steps: (i) draw a topic z from the multinomial distribution over topics $p(\mathbf{z}|R)$ in 2); (ii) draw a tag concept C_a from the corresponding distribution $p(C|z)$ in 1) over all tag concepts C.

However, the entities of interest in our work are tag concepts and we need to represent a tag concept in terms of the distribution of latent topics. This can be calculated by using the Bayes' rule with $p(C|z)$ and $p(z)$ as shown in Equation 3.1. The prior probability $p(z)$ has been mainly assumed as a uniform distribution in the previous literature [69, 71]. However, this often does not hold in real-world data. Therefore, we propose to use a non-uniform prior probability $p(z)$, computed as the ratio of the number of times that a particular topic z is assigned to any tokens in the Gibbs Sampling process, N_z, to the number of tokens in the whole resource collection, N, as shown in Equation 3.2. Finally, each tag concept can be represented as a $|\mathbf{z}|$-dimensional vector and the sum of the elements (probabilities) in the vector equal to 1 (see Equation 3.3, where $v(C)$ is the representation of a tag concept in terms of probabilistic distributions of latent topics).

$$p(z|C_a) \propto p(C_a|z) * p(z) \tag{3.1}$$

$$p(z) = \frac{N_z}{N} \tag{3.2}$$

$$v(C_a) = \{p(\mathbf{z}_i|C_a)\}_{i=1}^{|\mathbf{z}|} \tag{3.3}$$

It was noted earlier that a tag concept is assumed to be potentially ambiguous and might have complex meanings. This proposed representation intuitively captures the different meanings of a tag concept implied by the latent topics. Since a tag concept is usually only related to several topics, we introduce the notion of a *significant topic set*, $\mathbf{z}_a^{sig}$, which includes the latent topics whose value is above p, for tag concept C_a (see Equation 3.4). We set $|\mathbf{z}|$ as 1000 based on model perplexity (see Section 3.6.1.2) and p as 0.1 in this study[1].

$$\mathbf{z}_a^{sig} = \{z \mid z \in \mathbf{z} \text{ and } p(z|C_a) \geqslant p\} \tag{3.4}$$

3.4 Feature Generation

This section presents the feature generation process used quantify the degree that a concept is a hyponym of another given a context concept. The generated features form the input to the Classification and Testing module with respect to the experiments presented in Section 3.6.3. We believe that subsumption relations can be better established if we model the way how humans understand and interpret the meaning of tags. Three assumptions are proposed based on how humans determine subsumption relations. For two tag concepts C_a and C_b to have a subsumption relation:

Assumption 1. *Topic similarity - they must be similar to each other to some extent.*

The topic similarity (or dissimilarity) is calculated in the low dimensional semantic space.

Assumption 2. *Topic distribution - they should have topic distributions satisfying conditions on both topic coverage and focus.*

Intuitively, a hypernym and its hyponym should have overlapping topics. In terms of topic coverage, a hypernym should have a distribution spanning over more significant topics or dimensions than the hyponym. In terms of focus, the hyponym tends to have a high probability on one or few of the significant topics covered by the hypernym.

Assumption 3. *Probabilistic association - they should have a strong association to each other.*

Probabilistic association has its root in cognitive science and psychology [163]. It measures the degree of association between two concepts with a given context (e.g., parent of both concepts). In other words, it measures how likely that one is able to associate one concept given another and some background information. In our work,

[1]The value of p (= 0.1) is set empirically according to the distribution of $p(z|C_a)$ and the number of topics $|\mathbf{z}|$. For $|\mathbf{z}| = 1000$, the average $p(z|C_a)$ is 0.001, a very high p might produce no significant topics, while a very low p might include many irrelevant topics.

we quantify this likelihood using the conditional and joint probabilities of the two tag concepts.

Based on the above assumptions, we generate three corresponding categories of features that together characterise the degree of subsumption between pairs of tag concepts, as shown in Table 3.1 below.

TABLE 3.1: Feature sets corresponding to the three assumptions

Features	Description
	Topic Similarity Based Features
Cos_sim	Cosine similarity of two topic distribution vectors
KL_Div1	Kullback-Leibler Divergence from C_a to C_b
KL_Div2	Kullback-Leibler Divergence from C_b to C_a
Gen_Jaccard	Generalised Jaccard Index of two topic distribution vectors
	Topic Distribution Based Features
overlapping	Number of overlapping significant topics
diff_num_sig	Difference of the number of significant topics
diff_max	Difference of the maximum elements in two tag vectors
diff_aver_sig	Difference of the average probability of significant topics
	Probabilistic Association Features
$p(C_a\|C_b)$	Probabilistic association of C_a given C_b
$p(C_b\|C_a)$	Probabilistic association of C_b given C_a
$p(C_a\|C_b, R_{a,b})$	Local probabilistic association of C_a given C_b and $R_{a,b}$
$p(C_b\|C_a, R_{a,b})$	Local probabilistic association of C_b given C_a and $R_{a,b}$
$p(C_a, C_b)$	Joint probabilistic association of C_a and C_b
$p(C_a, C_b\|R_{a,b})$	Local joint probabilistic association of C_a and C_b given $R_{a,b}$

3.4.1 Topic Similarity Based Features

Assumption 1 is translated into several topic-based similarity and dissimilarity features. We use three distinct similarity measures, Cosine similarity, Kullback-Leibler Divergence and Generalised Jaccard Index.

3.4.1.1 Cosine Similarity

Cosine similarity, denoted as *Cos_sim*, is one of the most common similarity measures used in Information Retrieval [120, p. 110-p. 113] and Natural Language Processing [94]. The cosine similarity length-normalises the topic distribution vectors of two tag concepts to unit vectors, and computes their dot product, which is the cosine of the angle between the unit vectors.

$$\text{Cos\_Sim}(C_a, C_b) = \frac{v(C_a) \cdot v(C_b)}{|v(C_a)||v(C_b)|} \tag{3.5}$$

3.4.1.2 Kullback-Leibler Divergence

The Kullback-Leibler (KL) Divergence [100], or called relative entropy, expresses the difference between two probability distributions. Different from many other similarity

measures, KL Divergence is asymmetric, i.e. $D_{\mathrm{KL}}(P||Q)$ is different from $D_{\mathrm{KL}}(Q||P)$ for two distributions P and Q. In information theory and machine learning, $D_{\mathrm{KL}}(P||Q)$ measures the additional amount of information required when approximating P using Q instead of using the distribution P [16, p. 55]. When each concept is represented as a topic distribution, as elaborated in [190], it is the "surprise" received from one concept to another concept and this asymmetric "surprise" can imply the degree of subsumption between concepts: compared to $D_{\mathrm{KL}}(C_b||C_a)$, a higher value of $D_{\mathrm{KL}}(C_a||C_b)$ may imply much "surprise" received when we approximate the concept C_a with C_b, thus C_a is likely to be a hypernym of C_b. We thus generated two features, denoted as *KL_Div1* (or $D_{\mathrm{KL}}(C_a||C_b)$) and *KL_Div2* (or $D_{\mathrm{KL}}(C_b||C_a)$), respectively, defined in Equation 3.6.

$$D_{\mathrm{KL}}(C_a||C_b) = \sum_{i=1}^{T} v(C_a)_i \log \frac{v(C_a)_i}{v(C_b)_i} \tag{3.6}$$

3.4.1.3 Generalised Jaccard Index

Different from the other features in this category, the generalised Jaccard index, or the *fuzzy sets similarity* defined in the work [175], is based on the intersection and union of the topic sets between concepts, taking into consideration the magnitude of probability distributions. The metric can be regarded as a generalised version of the Jaccard Coefficient [171, p. 74] for real-valued vectors or probability distributions. The notion of sets from the (generalised) Jaccard Index matches well to the idea of measuring the concept similarity by their set of topics. The feature, denoted as *Gen_Jaccard*, is defined in Equation 3.7.

$$\mathrm{Gen\_Jaccard}(C_a, C_b) = \frac{\sum_i \min(v(C_a)_i, v(C_b)_i)}{\sum_i \max(v(C_a)_i, v(C_b)_i))} \tag{3.7}$$

3.4.2 Topic Distribution Based Features

The Assumption 2 is translated into the following features as shown in the second part of Table 3.1.

3.4.2.1 Number of Overlapping Significant Topics

Having overlapping significant topics is a simple while important indication of a subsumption relation. It is denoted as *overlapping* in Equation 3.8, where $\mathbf{z}_a^{\mathrm{sig}}$ and $\mathbf{z}_b^{\mathrm{sig}}$ can be obtained from Equation 3.4.

$$\mathrm{overlapping}(C_a, C_b) = |\mathbf{z}_a^{\mathrm{sig}} \cap \mathbf{z}_b^{\mathrm{sig}}| \tag{3.8}$$

3.4.2.2 Difference of the Number of Significant Topics

The number of significant topics is an indicator of how broad a tag concept is in terms of meanings. It is natural that general concepts tend to have more significant topics

than specific ones. The difference of the number of significant topics between C_a and C_b is used as a feature and is denoted as *diff_num_sig*, defined in Equation 3.9.

$$\text{diff\_num\_sig}(C_a, C_b) = |\mathbf{z}_a^{sig}| - |\mathbf{z}_b^{sig}| \tag{3.9}$$

3.4.2.3 Difference of Maximum Probability in Topic Distributions

The difference of the maximum probabilities given the two topic distributions is defined in Equation 3.10. This feature works jointly with *overlapping* and the topic similarity based features: If C_a and C_b are similar and share some overlapping topics, a positive value of this feature, $\text{diff\_max}(C_a, C_b)$, may imply that C_a is more specific than C_b. The intuition is that the maximum probability of a hyponym on a topic should be higher than that of the hypernym. We denote this feature as *diff_max* in the equation below, where $\max(v(C))$ returns the maximum entry in the probability distribution.

$$\text{diff\_max}(C_a, C_b) = \max(v(C_a)) - \max(v(C_b)) \tag{3.10}$$

3.4.2.4 Difference of the Average Probability of Significant Topics

The feature *diff_max* only captures the difference of maximum probabilities and is not enough for concepts which have multiple significant topics. We add another feature, the difference of the average probability of significant topics between C_a and C_b. It is calculated using Equation 3.11 and denoted as *diff_aver_sig*.

$$\begin{aligned} \text{diff\_aver\_sig}(C_a, C_b) &= \text{Aver}(\mathbf{z}_a^{sig}) - \text{Aver}(\mathbf{z}_b^{sig}) \\ &= \frac{\sum(\mathbf{z}_a^{sig})}{|\mathbf{z}_a^{sig}|} - \frac{\sum(\mathbf{z}_b^{sig})}{|\mathbf{z}_b^{sig}|} \end{aligned} \tag{3.11}$$

If $|\mathbf{z}_a^{sig}|$ or $|\mathbf{z}_b^{sig}|$ is zero, we set its corresponding average probability $\text{Aver}(\mathbf{z}_a^{sig})$ or $\text{Aver}(\mathbf{z}_b^{sig})$ as zero.

3.4.3 Probabilistic Association Based Features

The idea of probabilistic association among words is firstly proposed in [69, 71] and has its root in cognitive psychology [129]. It is believed that, in human memory, words have pre-existing associative structures constantly created from experiences [129]. With a probabilistic generative model, we can extract the gist of words and predict other associated ones based on bayesian inference [71]. We extend this idea and define new methods to quantify probabilistic associations among social tags under a given context.

The associative relations between words can be computed as a conditional probability of a response word given a cue word, marginalising over the hidden topics. While the *conditional probability* measures how likely one tag concept can be generated given another, the *joint probability* measures how likely two tag concepts can be generated

together. In addition, we introduce a third tag as the context for the computation, which can be the root concept of the domain or sub-domain under consideration, or the direct parent concept of the hypernym in the tag pair. This allows us to learn a concept hierarchy from top to bottom (see Section 3.5). As these features are extracted with a local context, they are referred to as *local* probabilistic associations. The six features in this category are summarised in the third part of Table 3.1 and described below.

3.4.3.1 Probabilistic Association

The probabilistic association between two tag concepts is defined as a conditional probability of one tag concept given another. The association is asymmetric and analogous to how we cognitively associate words [71]. The conditional probability $p(C_a|C_b)$ and $p(C_b|C_a)$ are computed by marginalising the inferred topics as shown in Equation 3.12.

$$\begin{aligned} p(C_a|C_b) &= \sum_{z \in \mathbf{z}} p(C_a|z, C_b) p(z|C_b) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) p(z|C_b) \end{aligned} \tag{3.12}$$

The $p(C_a|z)$ can be obtained from the LDA analysis in Section 3.3, and $p(z|C_b)$ can be obtained using Equation 3.1. We adopt the assumption made in [71] that C_a and C_b are conditionally independent given the latent topic z. Similarly, we can compute $p(C_b|C_a)$.

3.4.3.2 Local Probabilistic Association

When constructing a hierarchy using a top down approach, a potential subsumption relation between two tag concepts should be considered with respect to their common parent. The parent tag concept represents the local context under consideration, which would facilitate disambiguating the meanings of the two tag concepts. To capture this idea, we propose the concept of local probabilistic association, which is computed conditioned on a context tag $R_{a,b}$. It is asymmetric and we define two feature extraction functions, $p(C_a|C_b, R_{a,b})$ and $p(C_b|C_a, R_{a,b})$, as shown in Equation 3.13.

$$\begin{aligned} p(C_a|C_b, R_{a,b}) &= \sum_{z \in \mathbf{z}} p(C_a|z, C_b, R_{a,b}) p(z|C_b, R_{a,b}) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) p(z|C_b, R_{a,b}) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) \cdot \frac{p(C_b, R_{a,b}|z) p(z)}{p(C_b, R_{a,b})} \\ &= \sum_{z \in \mathbf{z}} \frac{p(C_a|z) p(C_b|z) p(R_{a,b}|z) p(z)}{p(C_b, R_{a,b})} \end{aligned} \tag{3.13}$$

Here we extend the assumption in [71] and assume that C_a, C_b and $R_{a,b}$ are conditionally independent given the latent topic z. The $p(C_a|z)$, $p(C_b|z)$, and $p(R_{a,b}|z)$ can be obtained from the LDA analysis; $p(z)$ is computed using Equation 3.2 and $p(C_b, R_{a,b})$ is computed by using Equation 3.14 (see Section 3.4.3.3).

3.4.3.3 Joint Probabilistic Association

Tag concepts that have a direct subsumption relation fall into similar areas and should have a high likelihood of being jointly generated. Therefore, we define the joint probabilistic association, $p(C_a, C_b)$. It is symmetric and computed using Equation 3.14, where $p(C_a|C_b)$ can be obtained by using Equation 3.12.

$$\begin{aligned} p(C_a, C_b) &= p(C_a|C_b)p(C_b) \\ &= p(C_a|C_b) \sum_{z \in \mathbf{z}} p(C_b|z)p(z) \end{aligned} \tag{3.14}$$

3.4.3.4 Local Joint Probabilistic Association

Similar to local probabilistic association, the local joint probabilistic association is further conditioned using a context tag $R_{a,b}$. It measures the likelihood of two tags being jointly generated with a particular context. It is also symmetric, denoted as $p(C_a, C_b|R_{a,b})$, where the $p(C_a|C_b, R_{a,b})$ and $p(C_b|R_{a,b})$ can be computed using Equations 3.12 and 3.13, respectively.

$$\begin{aligned} p(C_a, C_b|R_{a,b}) &= p(C_a|C_b, R_{a,b})p(C_b|R_{a,b}) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z, C_b, R_{a,b})p(z|C_b, R_{a,b})p(C_b|R_{a,b}) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) \cdot \frac{p(C_b, R_{a,b}|z)p(z)}{p(C_b, R_{a,b})} \cdot p(C_b|R_{a,b}) \end{aligned} \tag{3.15}$$

Once the three groups of features (14 features in total) are defined (see Table 3.1 for a summary), in the Classification and Testing module, we generate positive and negative instances, through tag grounding and instance labelling as described in Sections 3.6.2.1 and 3.6.2.2. Each instance is represented as a 14-dimensional feature vector. We create training, validation and testing datasets and feed the data into a classifier, which aims at learning a decision boundary in the feature space for binary prediction, i.e. whether subsumption relation holds between a new ordered pair of tag concepts given a context tag concept. The selection of classifiers is independent from our approach. We will test and evaluate several mainstream of-the-shelf classifiers in Section 3.6.3.

3.5 Hierarchy Generation Algorithm

Concept hierarchies, which can represent structured knowledge of a (sub-)domain, carry higher semantics than subsumption relations, according to the spectrum of structured knowledge in Figure 2.1. This section proposes a Hierarchy Generation Algorithm, in the Knowledge Enrichment module of the proposed machine learning system, to construct concept hierarchies, which are later used to enrich external KBs.

A hierarchy can be generated with an algorithm that organises tag concepts with valid subsumption relations from top to bottom, in an iterative manner. The algorithm starts with a specified "root" concept (a specific concept in a KB, which is designated by the users) and learns the layer below it. Then it learns the next layer from the current layer, and so on. The learned hierarchy is a Direct Acyclic Graph (DAG), where the nodes are tag concepts and edges represent subsumption relations among them. More specifically, it is a monohierarchy, where each concept can have at most one hypernym [194, p.140]. Such a hierarchy can be transformed to a formal lightweight, terminological ontology [63]. Usage of a monohierarchy is a common recommendation to construct ontologies and a strict requirement for classification systems [194, p.169, p.207], although in real world applications, one concept may have more than one hypernym, resulting in a polyhierarchy. We only aim to generate monohierarchies as a standard hierarchy; and polyhierarchies can be generated by relaxing the constraints on the number of parent concepts.

A key step in this algorithm is to select candidate hyponyms for a concept under consideration and then pass them to the trained classifiers for prediction. To enhance the consistency of the hierarchy generation, during the candidate hyponym selection, the algorithm makes use of the context of a concept, which is defined as the direct hypernym of that concept if available, otherwise, it is defined as the specified root concept. The candidate hyponyms of a concept should be associated to the concept, the root, as well as the context. The candidate selection condition is therefore calculated by using the global and local probabilistic association, according to Equations 3.12 and 3.13. Let *cand* be a candidate hyponym, *root* be the user-specified root concept, *concept* be the concept under consideration for which the candidate hyponyms are to be selected, *context* be the direct hypernym of *concept*, and TH be a pre-defined threshold. If the following two conditions are met then *cand* is chosen as a candidate hyponym of *concept*: (1) $p(cand|root) > TH$, this means that all candidates should be associated to the specified root; and (2) $p(cand|concept, context) > TH$, this means that all candidates should be associated to the concept under consideration given the context[2]. The two probabilities can be calculated based on the Equations 3.12 and 3.13, respectively.

The notations used in Algorithm 1 are explained as follows.

- G_{layer} represents a layer in the learned hierarchy; it is initialised as the root layer.

[2] TH is empirically set within $[\frac{1}{|C|}, \frac{10}{|C|}]$ for both conditions, where $|C|$ is the number of tag concepts. This is to ensure that TH is higher than the average probability while retaining a considerable number of candidates.

- H is the hierarchy to be generated; it is initialised as $\emptyset$.
- $h(x_i, \Theta)$ is the classification function to predict if a subsumption relation holds between two tag concepts (see Section 3.6.3). Θ represents the learned weights in training the classifier; $x_i = f(I_i)$ is an instance which is represented as a vector of the extracted features; and f represents the feature extraction function defined in Section 3.4.
- L is the list of associated tag concepts to the user specified root, i.e., $L \leftarrow \{cand \mid p(cand|root) > TH\}$. All the candidate hyponyms will be selected from this list.

When selecting the candidate hyponyms for the *root*, as *context* is not available, only the condition (1) is used (see line 2-4 in Algorithm 1). From line 5 to line 14, the algorithm learns the layer below the root. If the layer is not the root layer, then there are possibly multiple concepts on that layer. From line 16 to line 27, for each of the concepts, the algorithm selects a number of candidates from the list L. Then the pairs of each of the candidates and the concept under consideration are passed to the classification function h for prediction. If a subsumption relation can be established, then the pair is added into the temporary layer G'_{next}. The layer may need to be pruned and then added into the hierarchy H (lines 28-30, detail of the pruning process is presented in Algorithm 2). Then the algorithm learns the next layers with recursive calls (lines 31-33).

To create a monohierarchy, it is necessary to prune edges to ensure that each node (except the root) has only one hypernym. Algorithm 2 prunes a weighted directed graph with possible cycles. The input is an intermediate layer, G'_{next}, in Algorithm 1 and the output is G_{next}. The idea is to select the hypernym with the highest confidence score from the classification. In line 2, the algorithm first sorts the edges by their weights (i.e., classification scores) in descending order. In lines 3-8, for each edge E_i, it retrieves the hyponym *hypo*, which is then inserted if there is no parent for *hypo* in the G_{next} layer (function hasParent($hypo, G_{next}$) returns a boolean value).

The time-complexity of Algorithm 1 is $\mathcal{O}(d \cdot (l \cdot m \cdot c + m' \log m' + m'))$, where l is the number of possible candidate hyponyms; m and m' are the number of possible edges at the G_{layer} and G'_{next} respectively; d is the depth of the hierarchy H; and c is the time-complexity of the classifier function $h(x_i, \Theta)$. The graph pruning algorithm (Algorithm 2), as a part of Algorithm 1, has time complexity $\mathcal{O}(m' \log m' + m')$[3]. For most academic domains, the values of l, m, m', and d are limited; the time-complexity of the algorithm is dependent on the time-complexity c of the underlying classifier. Therefore, the algorithm is reasonably efficient. We further evaluate the algorithms and the quality of the learned structured knowledge in the experiment and evaluation.

[3]The sorting part (line 2 in Algorithm 2 uses Quicksort [82], adapted in the built-in function sort() in MATLAB for the implementation, having average time-complexity $\mathcal{O}(m' \log m')$.

Algorithm 1: generateHierarchy(G_{layer})

Require: G_{layer}, H, L, and h.
Ensure: H, hierarchy to be learned.

Initialise $G_{next} \leftarrow \emptyset$;
if G_{layer} *is the root layer* **then**
  Add *root* to H;
  $L \leftarrow \{cand \mid p(cand|root) > TH\}$;
  for *each cand in* L **do**
    $context \leftarrow root$;
    $I_i \leftarrow <cand, root, context>$;
    $x_i \leftarrow f(I_i) = [f_1(I_i), f_2(I_i), ..., f_{14}(I_i)]$;
    Predict subsumption relation using $h(x_i, \Theta)$;
    if *subsumption relation holds* **then**
      $G_{next} \leftarrow G_{next} \cup < cand, root >$;
      Remove *cand* from L;
    end
  end
else
  for *each edge* $< concept, context >$ *in* G_{layer} **do**
    $L_{sub} \leftarrow \{cand \mid p(cand|concept, context) > TH, cand \in L\}$;
    for *each cand in* L_{sub} **do**
      $I_i \leftarrow <cand, concept, context>$;
      $x_i \leftarrow f(I_i) = [f_1(I_i), f_2(I_i), ..., f_{14}(I_i)]$;
      Predict subsumption relation using $h(x_i, \Theta)$;
      if *subsumption relation holds* **then**
        $G_{next} \leftarrow G_{next} \cup < cand, concept >$;
        Remove *cand* from L;
      end
    end
  end
  $G_{next} \leftarrow \text{prune}(G'_{next})$;
end
Add G_{next} to H;
while *not finished* **do**
  generateHierarchy(G_{next})
end

3.6 Experiment and Evaluation

We conducted experiments using three large-scale, publicly available KBs, DBpedia, Microsoft Concept Graph (MCG), and ACM Computing Classification System (CCS). The training and testing data were automatically created by grounding the tag concepts in these KBs. The results were compared to those produced by the state-of-the-art mechanisms and evaluated using three strategies, i.e., relation-level, ontology-level and Knowledge Base Enrichment based evaluation. The implementation of the system and

Algorithm 2: prune(G'_{next})

Require: G'_{next}
Ensure: G_{next}, a pruned graph as a DAG.
Initialise G_{next};
Sort all edges ($E < hypo, hyper >$) in G'_{next} in descendant order by classification score;
for $i \leftarrow 1$ ***to*** $|E|$ **do**
  Retrieve the *hypo* from E_i;
  if $NOT\ hasParent(hypo, G_{next})$ **then**
    $G_{next} \leftarrow G_{next} \cup E_i < hypo, hyper >$;
  end
end

experiments are available on GitHub[4].

3.6.1 Social Tagging Data Processing

We extracted a social tagging dataset from Bibsonomy, which is a well-known social bookmarking system for academic publications and Web links, maintained by the Knowledge and Data Engineering Group at the University of Kassel [13]. We used the whole dump of the Bibsonomy data (version "2015-07-01"), which can be downloaded after request[5]. The whole dataset contains 3,794,882 annotations, 868,015 distinct resources and 283,858 distinct tags contributed by 11,103 users, accumulated from 2005 to July 2015.

3.6.1.1 Data Cleaning

To create a cleaned folksonomy $\mathbb{F}^{clean}$, we performed data cleaning steps as described in Section 3.2, including: (1) special character handling, based on the assumed meaning of special characters, we deleted the parts before ":" in a tag, separated a tag to multiple individual tags if it contained a comma (,), semicolon (;), slash (/) or brackets, and treated tags containing underscores (_) as a multiword tags; (2) multiword and single-word tag concept extraction (for single tags, a WordNet-based lemmatiser[6] was used); (3) tag filtering by metrics and languages; for example, we filtered out insignificant tags and only kept multi-word and single-word tag groups which have been used by no less than four distinct users. Also we only kept English tags based on the automatic detection results obtained using the Google Translation API[7]. Some examples of the data cleaning results are presented in Figure 3.2. We made openly available the cleaned multiword and single tag groups (or tag concepts) as a supplementary material of the

[4] `https://github.com/acadTags/tag-relation-learning/`
[5] `https://www.kde.cs.uni-kassel.de/bibsonomy/dumps/`
[6] `https://github.com/qingxiang-jia/lee-lemmatizer`
[7] `https://cloud.google.com/translate/`

paper [48][8]. For learning subsumption relations from tagging data, we further selected the resources as academic papers rather than websites in Bibsonomy (this cut almost half of the resources), and removed resources that have less than three tag concepts. Finally, we obtained a cleaned folksonomy of higher quality, $\mathbb{F}^{clean}$, with 7,458 tag concepts and 128,782 publications. Table 3.2 presents some statistics concerning both the raw and the cleaned Bibsonomy datasets.

Table 3.2: Statistics for the raw and the cleaned Bibsonomy dataset

	Tags/TCs	MTCs	STCs	Users	Res
Raw data	283,858	-	-	11,103	868,015
Cleaned data in [48]	17,379	2,502	14,877	-	663,148
Cleaned data (Res, as papers, $\geq$ 3 TCs)	7,458	2,293	5,165	-	128,782

Notes: TCs, Tag Concepts; MTCs, Multiword Tag Concepts; STCs, Single-word Tag Concepts; Res, Resources set R.

3.6.1.2 Probabilistic Topic Modelling from Tagging Data

Each resource was treated as a "bag of tags". For data representation and feature generation, Probabilistic Topic Modelling was performed with LDA and Gibbs Sampling by using the MALLET Machine Learning Library [121][9]. The two concentration parameters for the Dirichlet distribution in LDA [19, 185] were set empirically, according to [70] and the default settings in MALLET [121]: topic-word hyperparameter $\alpha = 50/|\mathbf{z}|$ [70]; and the document-topic hyperparameter $\beta = 0.01$ [121]. We held out 10% of the data to optimise the number of topics $|\mathbf{z}|$ with minimum perplexity and set $|\mathbf{z}|$ as 1000. We then used this probabilistic representation to extract features for learning.

Table 3.3: Example latent topics related to the tag concept "web"

Topic ID	**Most probable 5 tag concepts**
14	web accessibility centre mobility human
17	web mining web_mining data_mining data_web
126	web social social_web science web_science
247	semantic_web web semantic ontology rdf
333	application web web_application ajax web_interfaces
466	service web_service web composition service_composition
576	search web web_search social_search social_web
577	web archive crawl alexandria l3s

Table 3.3 provides an example on the learned topics, each of which is represented as a probabilistic distribution of tags. Only the five tag concepts with the highest probabilities in the distribution $p(C|z)$ are shown. It can be seen that collectively the tag concepts provide an intuitive definition on the meanings of the hidden topics. From a different perspective, probabilistic topic modelling is also an effective dimensionality

[8]The cleaned tag concepts and the detailed preprocessing steps can be acquired from `https://github.com/acadTags/tag-data-cleaning`

[9]`http://mallet.cs.umass.edu/`

reduction technique which transforms the original resource representation from a "bag of tags" to a vector of latent topics in a lower semantic space. A tag concept may relate to multiple topics, for example, the tag "web" is related to topics 14 (human accessibility), 17 (data mining), 126 (social Web) and 247 (semantic Web), 333 (Web applications), 466 (Web service), 576 (Web search), 577 (Web archiving and crawling). Tag concepts such as "web" contribute to multiple topics and are potentially general concepts. Then, we represent each tag as a distribution of the topics from $p(C|z)$ and $p(z)$, according to the Equations 3.1-3.3.

3.6.2 Labelled Dataset Creation

To learn subsumption relations, we need to generate labelled training and testing data. Selected tag pairs from the Bibsonomy dataset were automatically grounded to those in KBs and then labelled as either positive (subsumption) or negative.

3.6.2.1 Tag Grounding

Three external KBs were leveraged: (1) **DBpedia** contains structured information of Wikipedia, described in RDF (Resource Description Framework). We used the DBpedia "2015-10" version[10], to be consistent with the Bibsonomy dataset (2015 version). According to the ontological structure of DBpedia[11], we extracted concepts with subsumption relations using the *skos:broader* predicate and we used the *dbo:wikiPageRedirects* predicate to extract equivalent concepts to increase the recall of string matching; (2) **Microsoft Concept Graph (MCG)**[12] is a data-driven KB mined from billions of Web pages, released in September 2016, consisting of 85 million "is-a" relations and 18 million concepts. Each "is-a" relation is associated with a strength value. We selected the strength no less than 5, which resulted in 2.8 million relations; and (3) **ACM Computing Classification System (CCS)**[13] is an academic classification system that has been used to organise and retrieve publications by subjects in the ACM Digital Library. The lastest version (version 2012) was adopted in the experiments. From the RDF version of CCS, we treated *skos:broader* relations as subsumption relations and *skos:altLabel* as equivalent relations.

Table 3.4 provides some statistics concerning the concept overlap between external KBs and Bibsonomy. DBpedia had 2,191 common concepts with Bibsonomy and CCS had 691. The number is not excessive compared to the total number of tag concepts 7,458, suggesting that social tags can be potentially used to enrich human-engineered KBs. The number of overlapped concepts between MCG and Bibsonomy is 6,030, suggesting that there is still room to enrich the KB even though MCG is created from billions of Web pages.

[10] `http://downloads.dbpedia.org/2015-10/`

[11] For an example see the DBpedia Category, Machine Learning, `http://dbpedia.org/page/Category:Machine_learning`.

[12] `https://concept.research.microsoft.com/Home/Download`

[13] `https://www.acm.org/publications/class-2012`

TABLE 3.4: Statistics of the external Knowledge Bases (KBs) and the Bibsonomy folksonomy

	Concepts	Subsumption relations	Concept overlap with Bibsonomy	Release Date
DBpedia	1,316,674	2,706,685	2,191	2015-10
MCG	1,483,135	2,844,951	6,030	2016-09
CCS	9,060	2,390	691	2012 (latest version)
Bibsonomy	7,458	-	-	2015-07

3.6.2.2 Instance Labelling with Knowledge Bases

Supervised learning requires labelled data to train a model as complex function for prediction. We largely adapted and extended the approach in [144] to produce more balanced labelled data. We generated directed pairs of the overlapped tags concepts $< C_a, C_b >$, and labelled them with each KB. We used simple string matching, based on Levenshtein distance [109], to map a cleaned tag to a concept in the external KB. Then, a tag pair instance can be labelled as positive if there is an asserted, direct subsumption relation between the two tags in the external KB, and the probabilistic association between them, $p(C_a|C_b) > TH$, computed using Equation 3.12. This is to ensure the labelled instances are consistent with both the external KBs and Bibsonomy dataset. We created the negative instances by using the following methods: (i) reversed negative, for each positive pair $< C_a, C_b >$, we created a negative pair $< C_b, C_a >$; and (ii) random negative, if both randomly generated tag concepts appear in one of the KBs, but a subsumption relation between them cannot be found in any of the three KBs, we label the instance as negative. We also extracted the context tags for these instances to generate probabilistic association based features. Finally, we obtained 4,965 positive instances and 9,570 negative instances (including 4,785 reversed negative instances and 4,785 random negative instances). In total there are 14,535 instances and the ratio of positive to negative instances is around $1 : 1.93$.

It should be noted that the instance labelling process is based on the assumption that all relations in KBs are correct. In reality, the positive instances may suffer the quality issues of the KBs, as reported in [204] for DBpedia, due to the nature of the collaboratively generated data. Similarly, the random negative instances, according to the open-world assumption, may not necessarily be negative if they are not contained in any of the KBs. Nevertheless, the quality of these KBs is improving over time with the efforts of millions of individuals.

3.6.3 Classification Settings

Using the data created above, we generated features for each instance with the method proposed in Section 3.4 and fed them into different classifiers. We held out 20% of all instances for testing and used the remaining 80% for training. 10-fold cross-validation was used to tune the parameters and validate the generalisation of the trained models.

We used the standard precision, recall and F-measure to evaluate the performance of the classifiers. To test the effectiveness of the methods, we adopted four popular classification algorithms, namely, Support Vector Machine (SVM), AdaBoost, Logistic Regression and the CART algorithm (Classification And Regression Trees) [172, Chapter 3-4]. As each of the classification algorithms has its own characteristics and constraints, the evaluation was based on results from a group of classifiers, instead of any single classifier.

Support Vector Machine (SVM) is a maximum-margin classifier: it searches for a hyperplane which separates two classes with the maximum margin, where the margin is the perpendicular distance between two hyperplanes which touches the the closest data items in each class. With a kernel trick that transforms the original coordinate space of the data, SVM can be used to create nonlinear decision boundaries. Using a soft-margin approach, SVM can tolerate small number of training errors to form decision boundaries with better generalisability. SVM also has strong regularisation capabilities with its hyper-parameters, being able to control the complexity of the model to achieve good generalisation performance and thus to prevent from overfitting. AdaBoost (short for Adaptive Boosting) is a typical boosting algorithm for *ensemble learning*, which provides a structure to improve performance by aggregating the prediction of multiple normal weak classifiers. The weak classifiers are selected by re-sampling the training data based on the weight for each data item, and the weights are iteratively updated. Data items which are wrongly classified will have higher weights in the next iteration, thus AdaBoost is insusceptible to overfitting. Logistic Regression is a *generalised regression model* for categorical values (two categories in our case) adapted from linear regression. Finally, CART is a *decision tree learning* algorithm that aims that searching for a hierarchical structure where non-terminal nodes represent features, leaf nodes as class labels and edges as logical paths to generalise the data items; an impurity measure or split criteria is used to determine the goodness of partition. Detailed introduction of these classification algorithms can be founded in the book [172, Chapter 3-4].

We used the LibSVM 3.22[14] [30] Matlab version for SVM training. We used the radial basis function (RBF) kernel for SVM, and tuned the two parameters c and γ with grid-search to optimise the F_1 score, as suggested in [87]. The remaining three algorithms (CART, Logistic Regression and AdaBoost) were implemented in the Classification Learner App[15] in Matlab. We set the number of weaker learners as 30 and each of them used the same settings as the CART algorithm, and a shrinkage learning rate was set to 0.1 to prevent overfitting. All models were trained and validated using 10-fold cross-validation.

3.6.4 Evaluation

Three strategies were used for the evaluation: (i) relation-level evaluation using the testing set; (ii) ontology-level evaluation using external KBs as the gold standard; and (iii)

[14] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[15] `https://cn.mathworks.com/help/stats/classification-learner-app.html`

Knowledge Base Enrichment based evaluation through human assessment. The results allowed us to see to what extent social media data can be exploited to learn structured knowledge to enrich existing KBs. Finally, we visualise some learned hierarchies for analysis.

3.6.4.1 Relation-level Evaluation

We compared the performance of the proposed method to several representative studies as explained in the following. The feature set proposed in this work is denoted as FS_{all}, which consists of features related to topic similarity ($FS_{topicSim}$), topic distribution ($FS_{topicDist}$) and probabilistic association ($FS_{probAsso}$) (see the whole three feature sets in Table 3.1).

1. Binary classification using co-occurrence related features [144]: Combining several heuristics as features in previous studies, i.e., support and confidence [150], cosine similarity of tag-tag vector in [95, p. 56-59] and [160], set inclusion and generalisation degrees [123], mutual overlapping [28] and graph-based taxonomy search [80]. In total there are 8 features and the feature set is denoted as FS_{co}.

2. The method in [190] based on *Information Theory Principle for Concept Relationship*: This proposed two conditions to measure the degree of subsumption between two concepts. The first condition is the *similarity condition*, measuring the similarity between two concepts; the second condition is the *divergence difference condition*, which calculates the difference between the Kullback-Leibler divergence of two tag concepts. This is generally equivalent to the topic similarity based feature set in our method. It contains 4 features, denoted as $FS_{topicSim}$.

3. The topic distribution related features, $FS_{topicDist}$: To allow performance comparison with using only the topic distribution.

4. The probabilistic association features, $FS_{probAsso}$: To allow performance comparison with using only the probabilistic association.

5. Combining both the co-occurrence related features [144] and the feature sets proposed in this study: To determine if the performance of the proposed method can be further improved by combining the co-occurrence based features. In total there are 22 features, denoted as FS_{all}+FS_{co}.

The results are presented in Table 3.5. In general, using the feature sets FS_{all} achieved higher F_1 scores with a large margin than using any others, and the best ranking (ranked first with SVM and Adaboost and second with LR and CART). The performance was stable and consistent with different classification techniques, showing the robustness of the proposed feature set in characterising subsumption relations. Co-occurrence based features (FS_{co}), which have achieved impressive results for supervised learning, as reported in the study presented in [144], did not perform well for with

TABLE 3.5: Classification testing results with comparison among feature sets

Feature set	Classifier	Recall	Precision	F_1 score
Full feature sets, FS_{all}	SVM RBF($2^{10.5}$,$2^{4.5}$)	51.56 (1)	52.95 (3)	52.25 (1)
	AdaBoost	50.15 (1)	63.52 (3)	**56.05** (1)
	LR	34.04 (2)	65.00 (2)	44.68 (2)
	CART	45.02 (3)	62.87 (2)	52.46 (2)
Rêgo *et al.* [144] (co-occurrence features FS_{co}, including [95, 150], [28, 80, 123, 160])	SVM RBF(2^{10},2^{7})	36.96 (5)	58.81 (2)	**45.39** (4)
	AdaBoost	27.49 (4)	61.07 (4)	37.92 (4)
	LR	19.64 (3)	56.20 (4)	29.10 (3)
	CART	27.19 (4)	58.95 (4)	37.22 (4)
Wang *et al.* [190] (based on $FS_{topicSim}$)	SVM RBF($2^{10.5}$,2^{9})	46.02 (3)	47.02 (5)	**46.51** (3)
	AdaBoost	17.52 (5)	59.59 (5)	27.08 (5)
	LR	15.01 (4)	54.78 (6)	23.56 (4)
	CART	11.78 (5)	66.10 (1)	20.00 (5)
Topic distribution, $FS_{topicDist}$	SVM RBF(2^{10},2^{11})	40.28 (4)	46.14 (6)	**43.01** (5)
	AdaBoost	11.48 (6)	59.07 (6)	19.22 (6)
	LR	10.27 (6)	55.14 (5)	17.32 (6)
	CART	3.02 (6)	47.62 (6)	5.68 (6)
Probabilistic association, $FS_{probAsso}$	SVM RBF(2^{12},$2^{8.5}$)	27.80 (6)	60.53 (1)	38.10 (6)
	AdaBoost	44.51 (3)	63.60 (2)	52.37 (3)
	LR	14.20 (5)	68.12 (1)	23.50 (5)
	CART	53.07 (1)	60.09 (3)	**56.36** (1)
Combining full features with co-occurrence features in [144], FS_{all}+FS_{co}	SVM RBF($2^{9.5}$,2^{4})	49.25 (2)	52.41 (4)	50.78 (2)
	AdaBoost	46.32 (2)	65.25 (1)	**54.18** (2)
	LR	36.56 (1)	62.69 (3)	46.18 (1)
	CART	46.73 (2)	57.35 (5)	51.50 (3)

The values (2^a,2^b) after SVM RBF are the parameters c and γ tuned to optimise F_1 score. The highest F_1 score for each feature set is bolded. The number in all brackets shows ranking of the feature set under the same classifier.

our large labelled dataset in the academic domain. F_1 scores obtained using the co-occurrence based features (FS_{co}) [144] were much lower compared to FS_{all} (absolutely lower by 6.86% with SVM and by 18.13% with AdaBoost). This is probably because the Bibsonomy data is sparse, thus, many subsumption relations between low frequent tags are fail to be captured by data co-occurrences. Adding the co-occurrence based features (FS_{co}) to the proposed features sets, FS_{all}+FS_{co}, did not improve performance, showing that data co-occurrence does not provide further information to the proposed feature sets based on probabilistic topic modelling.

We also compared the proposed method to [190], which applied probabilistic topic modelling on a collection of scientific publication abstracts and then detected subsumption relations with the *Information Theory Principle for Concept Relationship*. The approach can be adapted into the supervised learning setting that uses the topic similarity features $FS_{topicSim}$. The proposed features FS_{all} performed better in terms of all metrics (in terms of F_1, an absolute increase by 5.74% with SVM and by 28.97% with AdaBoost). One of the main reasons is that the dataset used in [190] contains texts and rich contextual information, which is not the case for social tagging data.

When using the single feature set we found that probabilistic association ($FS_{probAsso}$) generated higher precision (overall best ranking), while the recall was lower than others. The best performance was achieved by using the full feature sets. This confirms the hypothesis that we can better characterise subsumption relations through all the feature sets founded on the three assumptions. We noticed that classification with $FS_{probAsso}$ and CART obtained a slightly higher F_1 score (+0.3%) than FS_{all} and Adaboost (56.34% vs. 56.05%), with the former having higher recall (+2.92%) but lower precision (-3.43%). The performance with CART was, however, not consistent with other classifiers and the overall ranking of the $FS_{probAsso}$ was worse than FS_{all}. This is probably because the individual features in $FS_{probAsso}$ can better satisfy the impurity criteria and are suitable for the rectilinear decision boundaries of the CART algorithm [172, p. 143-p. 147], while the other features which have strong interactions among them, especially those in $FS_{topicDist}$ (only 5.68% F_1 with CART but 43.01% with SVM), are more suitable for models with nonlinear boundaries and better generalisation capabilities. SVM and AdaBoost performed generally better than Logistic Regression (LR) and CART within each feature set. It is also noticed that, compared to the other 3 classifiers, training the SVM models with grid search to find the best parameters is computationally expensive, e.g., with best c values varying from $2^{9.5}$ to 2^{12} and γ values from 2^4 to 2^{11} as shown in Table 3.5.

The relation-level evaluation above shows that the proposed feature sets best characterise the subsumption relations between tags and achieved overall best prediction with the classifiers. It is also necessary to test the quality of the concept hierarchies, which captures knowledge in a domain or a sub-domain, with higher semantics than subsumption relations, generated by the proposed algorithm.

3.6.4.2 Ontology-level Evaluation

The ontology-level evaluation was designed to measure the quality of the hierarchies (or lightweight ontologies) derived using the hierarchy generation algorithm. We used a reference-based strategy adopted from the study in [165]. The prerequisite of this strategy is the existence of a "gold-standard" ontology to be compared against. The quality of the learned hierarchies is thus measured as the similarity to the "gold standard". This automated evaluation can ensure reproducibility, compared to manual assessment of concept hierarchies. We chose the popular KBs, DBpedia and CCS as the "gold standard" and aimed to test the capabilities of classifiers and the algorithm for generating hierarchies, although we are aware of the fact that both KBs are not perfect and the CCS has been relatively static (last updated 7 years ago at the time of writing this thesis). The data-driven KB, MCG, is not chosen as a "gold standard", because the transitivity of subsumption relations in MCG (which is an acyclic graph and suffers from semantic drift) is low [112].

We adopted the standard metrics for reference-based evaluation, *taxonomic precision* (TP), *taxonomic recall* (TR), *taxonomic F-measure* (TF) [43] and *taxonomic overlapping*

(TO) [119], also applied in [165]. The idea is to find a common concept C_c between a learned hierarchy L and a referenced hierarchy G, and to extract a *characteristic extract* from each of them, $\mathrm{ce}(C_c, L)$ and $\mathrm{ce}(C_c, G)$. The characteristic extract is defined as the *common semantic cotopy*, i.e., starting from the concept C_c to traverse the hierarchy L (or G) to find all the super- and sub-concepts of C_c in this hierarchy (except C_c itself) which are also presented in the other hierarchy G (or L), introduced in [43] and [42, p. 18]. The partial similarity of the two characteristic extracts regarding the common concept C_c is then calculated. The local taxonomic precision and recall regarding the common concept C_c can be calculated using Equations 3.16 and 3.17.

$$tp(C_c, L, G) = \frac{|\mathrm{ce}(C_c, L) \cap \mathrm{ce}(C_c, G)|}{|\mathrm{ce}(C_c, L)|} \tag{3.16}$$

$$tr(C_c, L, G) = \frac{|\mathrm{ce}(C_c, L) \cap \mathrm{ce}(C_c, G)|}{|\mathrm{ce}(C_c, G)|} \tag{3.17}$$

The global taxonomic precision $TP(L, G)$ and recall $TR(L, G)$ are computed by averaging all local tp and tr with respect to all common concepts. The taxonomic F-measure is the harmonic mean of both taxonomic precision and recall.

$$TP(L, G) = \frac{1}{|L \cap G|} \sum_{C_c \in L \cap G} tp(C_c, L, G) \tag{3.18}$$

Taxonomic overlapping is symmetric and can be used independently. The local version is defined as follows and the global version $TO(L, G)$ is computed by averaging all the local ones.

$$to_{\mathrm{ce}}(c, L, G) = \frac{|\mathrm{ce}(c, L, G) \cap \mathrm{ce}(c, G, L)|}{|\mathrm{ce}(c, L, G)| \cup \mathrm{ce}(c, G, L)|} \tag{3.19}$$

$$TO(L, G) = \frac{1}{|L \cap G|} \sum_{c \in L \cap G} to_{\mathrm{ce}}(c, L, G) \tag{3.20}$$

We used several domains in external KBs for ontology-level evaluation. For DBpedia, concepts matched to those within the top 5 layers under the categories "Areas_of_computer_science" and "Information_science" were selected (the domain is denoted as "CS/IS"). For the domains of "Education" and "Economics", concepts within the top 3 layers were selected. For CCS, all tag concepts matched to the uppermost 2, 3 or 4 layers were selected. We finally obtained 217 tag concepts in CS/IS, 226 in Education and 152 in Economics in DBpedia, and 43, 113, 133 tag concepts matched to the uppermost 2, 3, 4 layers of CCS, respectively. For each tag concept in the selected domain, we generated a sub-hierarchy using the hierarchy generation algorithm and calculated TP, TR, TF and TO (averaged results over the sub-hierarchies for each domain are reported). This novel evaluation process on multiple hierarchies is more rational than on only one global hierarchy against the KBs. The latter approach may be biased as it does not test the similarity of the branches between two hierarchies [165].

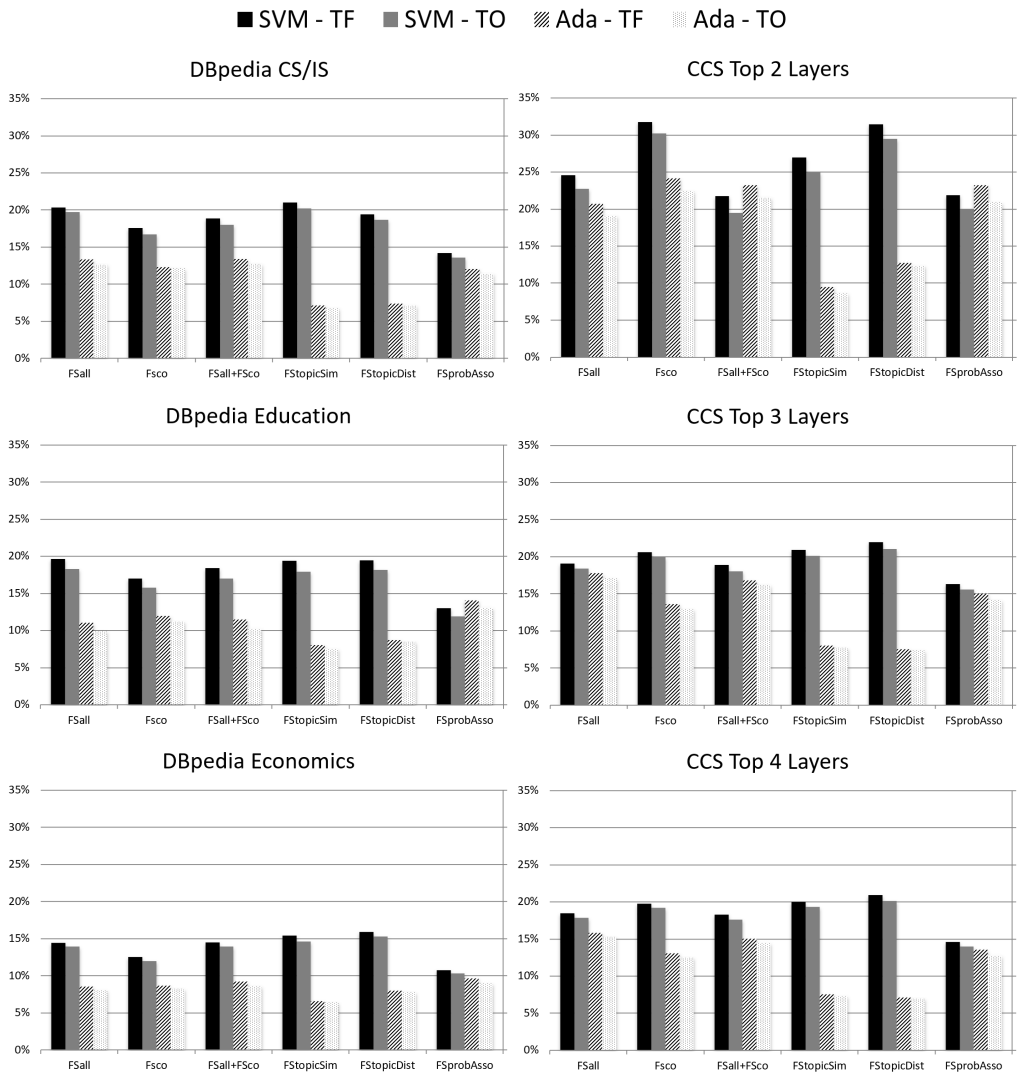


FIGURE 3.3: Results of ontology-level evaluation. The figures show the TF and TO values computed with the learned hierarchies from the Bibsonomy dataset and the "gold standard" (DBpedia and CCS). Three domains were selected for DBpedia, Computer Science/Information Science, Education and Economics; and three sub-hierarchies uppermost 2, 3 and 4 layers were tested for CCS. SVM or AdaBoost (denoted as "Ada") were used for classification. The x-axis represents methods with different feature sets and the y-axis represents the similarity in percentage. Higher TF and TO values indicate greater similarity to the gold standard.

Figure 3.3 shows the results obtained with different combinations of KBs, features sets and classifiers. The results demonstrate satisfying description ability of the proposed feature sets with the hierarchy generation algorithm, with generally better and more consistent results compare to other feature sets. The TF and TO values are also consistent with those reported in the previous study [14]. In all three domains of DBpedia, the TF and TO scores generated with the proposed features FS_{all} were generally higher than those generated with other features sets based on co-occurrence, topic similarity, topic distribution and probabilistic association. There were few exceptions, however, their performance was highly inconsistent between classifiers, e.g. the topic similarity

features $FS_{topicSim}$ had higher TF than FS_{all} for CS/IS using SVM, but much lower TF using Adaboost. For CCS with 2 uppermost layers, the highest TF and TO scores were obtained with the co-occurrence-based features, but the proposed feature set performed generally better with concepts matched to 3 and 4 uppermost layers, especially using Adaboost. This shows the advantage of the proposed feature set on generating hierarchies with more *specific* concepts than the co-occurrence-based features. Furthermore, results of the proposed feature set with CCS were also consistent between classifiers. Similar to the results in the relation-level evaluation, the performance of using only the topic similarity or topic distribution based features varied significantly with different classification techniques in all settings.

The ontology-level evaluation above shows that proposed feature sets with SVM and Adaboost can produce comparable and more consistent results based on similarity to existing KBs. This evaluation strategy, however, cannot judge the new concepts and relations, i.e. those not captured in existing KBs; we thus introduce Knowledge Base Enrichment based Evaluation.

3.6.4.3 Knowledge Base Enrichment Based Evaluation

One particularly interesting part of this research is to discover previously unseen knowledge or emerging semantics from social tagging data. The enrichment-based evaluation is to assess to what extent the method can enrich external KBs with new and meaningful concepts and relations. For this purpose domain experts were used for manual assessment.

We selected a number of concepts from DBpedia (the "CS/IS" domain) and CCS (the 2 uppermost layers) and used the trained classification models to predict their direct hyponyms. Then we identified new hyponyms which do not appear in the "gold-standard" KBs and let the human experts make judgement about their validity. To identify *new* relations, we controlled the release date as a factor, both external KBs, DBpedia and CCS are released or adopted slightly later than the social tagging data Bibsonomy, see Table 3.2: we used the "2015-10" version of DBpedia is about the same time (3 months later) to the Bibsonomy version (2015-07); for CCS, the release date is in 2012, but it is the lastest version, and still using as the backbone of the ACM digital library during this study. The Knowledge Base, MCG, is not used for this evaluation, as it has a low transitivity among the concepts and thus we were not able to select a specific domain concept hierarchy from MCG. A large number of direct subsumption relations was generated and around 99% of them were not present in the two existing KBs, DBpedia or CCS; in total, there were 3,846 distinct new relations for DBpedia, and 1,302 for CCS, based on the predictions from the SVM and AdaBoost models, as shown in Table 3.6. The overlapped relations between the learned structured knowledge and DBpedia or CCS are very few, which can be partly explained by earlier conclusions as (i) different annotation process between social tagging and the KBs [105]; (ii) different domain coverage and semantic granularity of concepts between the social tags and the

Table 3.6: Statistic of Knowledge Enrichment from folksonomies

	DBpedia		CCS	
	Enriched	Overlapped	Enriched	Overlapped
SVM	2876	34	890	3
AdaBoost	2079	17	944	3
Distinct Total	3846	36	1302	5

KBs [7, 33], and is also due to (iii) the various forms of concepts or terminologies from social media users and knowledge engineering experts.

As the number of enriched relations is large, we selected a subset (298 out of 5,148) for manual assessment based on the classifiers' confidence score of the learned relations from the prediction: we set the confidence threshold TH_c as 6 for SVM and as 0.6 for AdaBoost to narrow the list of relations to be manually assessed. Thirteen domain experts, including four academic staff members and nine senior PhD candidates, from universities in the UK and the US, were invited to participate the evaluation. They work in different areas of computer or information sciences. In the evaluation sheet, we asked them to mark the predicted relations with one of the four options:

- (subsumption) C_a is a narrower concept of C_b given C_r.
- (related) C_a is not a narrower concept of C_b, but they are related concepts.
- (unrelated) Nor the subsumption or related relation holds for the two concepts.
- (unsure) The participant is not sure about the answer.

Using the proposed method with SVM and AdaBoost, we generated two sets of subsumption relations for DBpedia and CCS respectively. We merged the results in the evaluation sheet and ended up with 298 distinct relations after filtering out those with low confidence scores. The multi-rater Fless Kappa [55] was 0.15 and free-marginal kappa [140] was 0.22 among the domain experts, showing a "slight" agreement. This is also consistent with the results reported in previous studies, e.g., Fless Kappa 0.137 in [59] and free-marginal kappa 0.139 in [168]. This "slight" agreement is because that the learned relations and concepts concern the very specific sub-areas and rare topics, thus some of them (especially abbreviations) may not be familiar to all participants.

Among the 3,874 ratings (298×13) presented to the judges, 1,489 of them (38.44%) were marked as "subsumption", and 1,131 (29.20%) were "related". We further compared the enrichment accuracy in terms of KBs. The ground truth was determined by assuming no less than a certain number of votes were for "subsumption" and the accuracy was computed with respect to the ground truth. As shown in Figure 3.4, the x-axis represents settings for the classifiers and KBs, and the y-axis represents the accuracy of the enriched relations. If we define a predicted relation as a true subsumption when at least five domain experts have the agreement, then the overall accuracy of the enriched relations was 53.36%. The accuracy increased to 66.44% and 74.50% if we only need

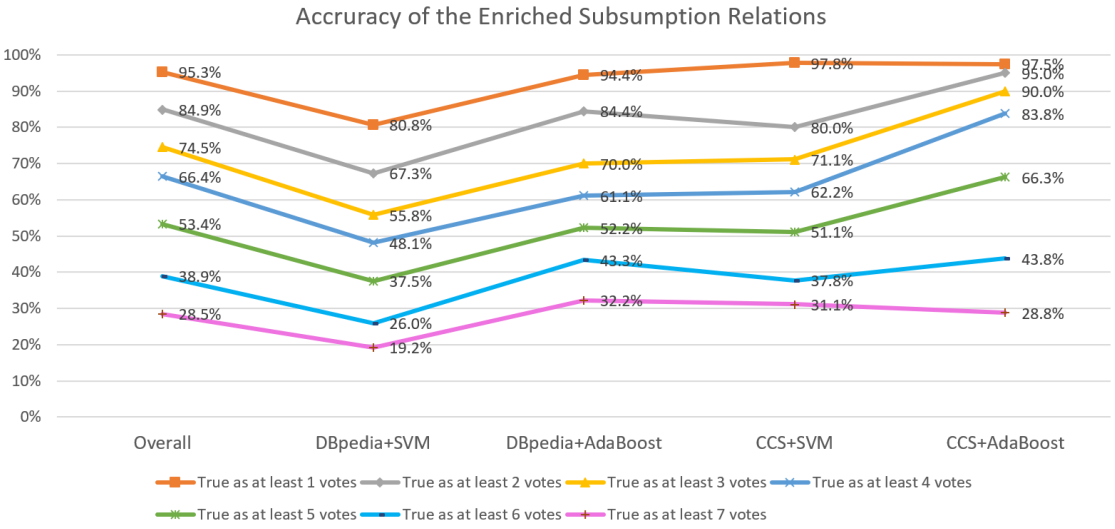


FIGURE 3.4: Results on Knowledge Base Enrichment based evaluation

agreement from four and three domain experts respectively; the accuracy decreased to 28.52% when we need agreement from seven of them. Higher accuracy was seen in most cases when enriching CCS than DBpedia. The reason might be that the concepts in CCS (2 uppermost layers) are more general and the hierarchy is more shallow than those of DBpedia. Therefore, there is much room for new relations and concepts in the selected layers of CCS. The results clearly show that the proposed method can help discover meaningful knowledge from noisy tagging data[16].

3.6.4.4 Hierarchy Visualisation

Finally, we present some of the learned hierarchies in Figure 3.5 and 3.6, to enrich the "data_mining" hierarchy in DBpedia and the "social_software" hierarchy in CCS using the proposed whole feature set, predicted with SVM and AdaBoost respectively. More learned hierarchies, also including "research_methods", "machine_learning", "information_retrieval" hierarchies to enrich CCS and an "e_commerce" hierarchy to enrich DBpedia, are presented in the Appendix A. Due to the size limitation, we used the "force-directed" layout[17] [56] to visualise the learned hierarchies as DAGs. The right arrow → points from a hypernym (broad concept) to a hyponym (narrow concept). It can be seen from the hierarchies that the concepts and relations from user-generated data present distinct terminology from structured knowledge created by domain experts and knowledge engineers. Also, many learned relations are reasonable, for example, "data_mining → association_rules", and "social_software → second_life". There are also "strange" relations, which still can reflect users' perspective or the bias and noise in the tagging data, for example, "data_mining → tobuy", which may indicate an application

[16]The evaluation sheet and the ratings from the domain experts are available on `https://github.com/acadTags/tag-relation-learning`

[17]The layout settings are available as a built-in parameter in MATLAB since version 2015B, see `https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.graphplot.layout.html#buxdj61-method`

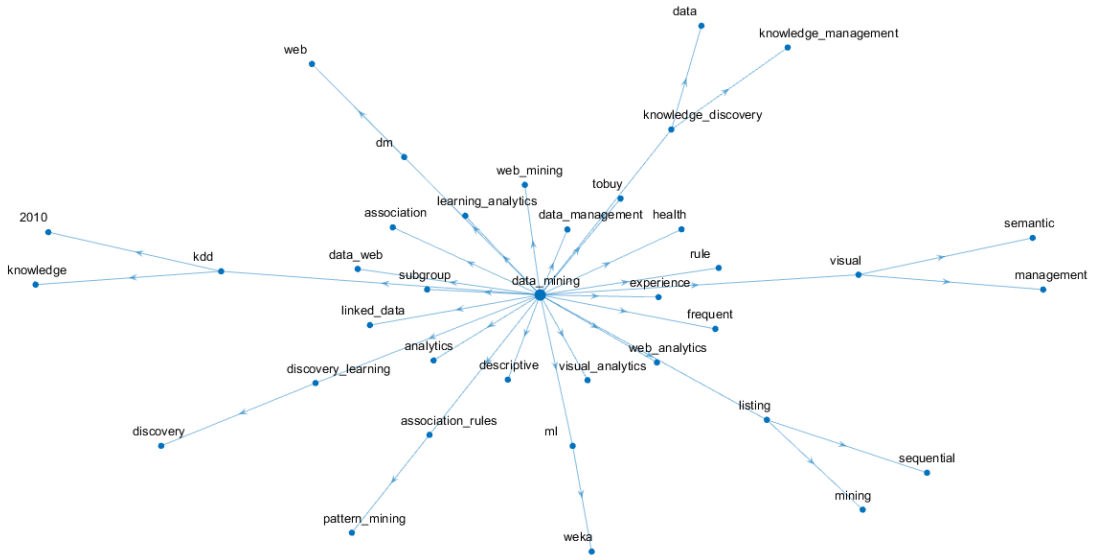


FIGURE 3.5: Excerpt of the learned hierarchy to enrich DBpedia in the domain of *data mining*, trained with the proposed full feature set FS_{all} using SVM.

area of data mining; some relation may not be strictly subsumption relations, but association relations, such as "social_software → web_2.0", consistent with the results from Knowledge Base Enrichment based evaluation that around 30% are marked as "related". The learned hierarchies have mostly no more than 3 layers, controlled by the Hierarchical Generation Algorithm to keep the consistency of concepts in the hierarchy. One limitation of the learned hierarchies is that the concepts on the same layer (or siblings) are not consistent to each other in terms of semantics, i.e. "kdd", "linked_data" and "subgroup". This consistency may be improved through an algorithm to find a global optimal hierarchy, instead of the greedy-based algorithm for hierarchy generation.

3.7 Related Work

In this section, we relate the proposed system to previous work on learning structured knowledge from social tagging data, mainly presented in Section 2.3. Current approaches in the literature can be categorised into several types: heuristics-based methods [80, 124], semantic grounding to external resources based methods [47, 59], unsupervised learning based methods [190, 211], and supervised learning based methods [144]. The proposed machine learning system is related to all the categories, but focuses on binary classification in supervised learning as in [144]. Distinct to [144], our approach utilises features extracted using an unsupervised learning method, Probabilistic Topic Modelling, inspired by the study in [190]; also the semantic grounding process for instance labelling is further extended based on three large human-engineered or data-driven KBs. The proposed feature set, founded on three assumptions to characterise the subsumption relations, can better quantify the relations from sparse tagging data than co-occurrence-based heuristics, especially for narrow folksonomies [183] in the academic domain. The

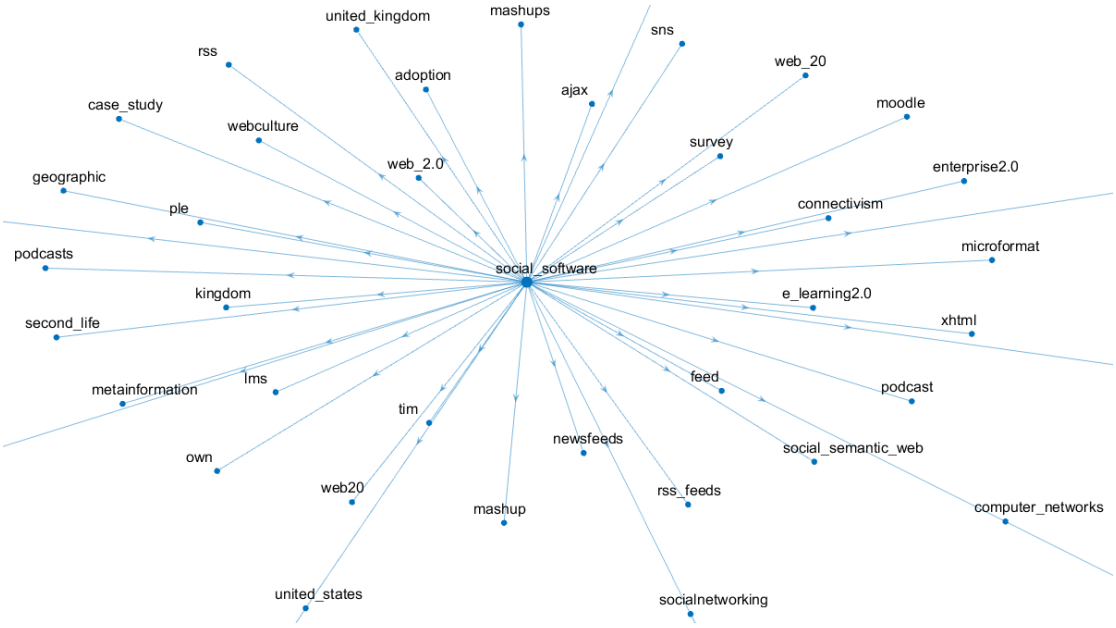


FIGURE 3.6: Excerpt of the learned hierarchy to enrich CCS in the domain of *social software*, trained with the proposed full feature set FS_{all} using AdaBoost.

method of the machine learning system finds an equilibrium among the methods, taking advantage of all the categories of approach. The proposed feature sets and the Hierarchy Generation Algorithm have been thoroughly evaluated in the experiment.

In terms of evaluation of knowledge derived from folksonomies, the most comprehensive evaluation study prior to our work is in [165]. The study [165] applied reference-based evaluation (similar to ontology-level evaluation), manual evaluation (on the relation-level) and pragmatic evaluation (to simulate navigation based on the hierarchy) to compare the unsupervised, clustering-based approaches to heuristic, graph-based approaches to learn hierarchies from tags. The study in this thesis evaluated supervised learning approaches, through automated relation-level and ontology-level evaluation, which were not evaluated in [165]. Besides, the proposed evaluation strategies further focused on Knowledge Base Enrichment based Evaluation, which explores how user-generated social media data can enrich existing KBs; this aspect of evaluation has not been considered in previous studies.

Regarding the Knowledge Base Enrichment, reviewed in 2.3.5, few prior studies have explored this direction despite the rich semantics from folksonomies over professional taxonomies and thesaurus [105, 116]. The most representative work for Knowledge Base Enrichment from folksonomies is [5], which proposed a system pipeline, "3E" techniques (Extraction, Enrichment, Evolution), to process and enrich the knowledge from tags to KBs. This is generally related to the modules in the proposed machine learning system: "Extraction" can correspond to Data Cleaning module, "Enrichment" to semantic grounding, and "Evolution" to the Knowledge Enrichment module. However, the study most focuses on extracting tag concepts and co-occurrence relations, instead of on learning subsumption relations or concept hierarchies, to enrich KBs. Much human

intervention with visualisation techniques was required for Knowledge Enrichment in [5]; this is also the case for Knowledge Base Enrichment from learners' tags in an e-learning environment in [60]. Our approach focuses on enriching KBs with structured knowledge of higher semantics, e.g. concept hierarchies, with reduced human intervention.

3.8 Summary and Discussion

In this chapter, we have introduced a novel machine learning system to learn structured knowledge from social tagging data. The system inputs raw social tagging data and external KBs, and outputs subsumption relations and concept hierarchies derived from the noisy, ambiguous, and flat tagging data. From the evaluation results and the visualised hierarchies, we can conclude that new structured knowledge of sufficient quality can be derived to enrich KBs. Many relations in the learned concept hierarchies from tags are previously unseen in the existing KBs. This validates the main hypothesis of this research regarding the user-generated social media data as "collective intelligence", i.e. a viable source to derive structured knowledge.

The novelty of the method lies in a supervised learning framework with training data automatically extracted based on probabilistic topic modelling, and a subsequent Hierarchy Generation Algorithm utilising the machine learning model. The system includes five connected modules. The *Data Cleaning* module extracts tag concepts from the raw, noisy and sparse tag sets. The *Data Representation* module, based on probabilistic topic modelling, represents the ambiguous meaning of each tag concept as a distribution of topics and reduces the dimensionality of tag concepts. The *Feature Generation* module further quantifies subsumption relations, founded on three assumptions based on topic similarity, topic distribution, and probabilistic association to form the features. The *Classification and Testing* module generates labelled data with external KBs and learns classifier models to predict new relations. The final *Knowledge Enrichment* module generates tag concept hierarchies with the learned classifier models and enriches existing KBs. In the system, the issues of noisiness, ambiguity, and sparsity of tagging data are addressed and the flat, unstructured tags are transformed into structured forms.

There is a lack of studies to formally evaluate the learned structured knowledge from social tagging data, especially in terms of Knowledge Base Enrichment. Thus a comprehensive evaluation was carried out towards the quality of the discovery knowledge using three different strategies, relation-level evaluation, ontology-level evaluation, and Knowledge Base Enrichment based manual evaluation, using the publicly available Bibsonomy dataset and three popular, human-engineered or data-driven KBs. The relation-level evaluation shows the proposed feature set based on probabilistic topic modelling better characterises subsumption relation between tags over co-occurrence-based features [144] and topic similarity-based features in [190]. The ontology-level evaluation demonstrates the competitive and consistent description ability of the feature set and the usefulness of the Hierarchy Generation Algorithm. The Knowledge Base Enrichment based evaluation

clearly shows that existing KBs created by domain experts and knowledge engineers can be supplemented by knowledge learned from the user-generated social media data. Hierarchy visualisation further shows that the learned concept hierarchies are meaningful and distinct from the existing knowledge sources.

The overall machine learning system is not without its conditions and limitations. The probabilistic topic modelling based representation requires sufficient amount of textual data to infer the hidden topic structure. The instance labelling with KBs requires sufficient coverage of the tags in the KBs, and assumes that the concepts and relations in the KBs are correct and fairly comprehensive. The system requires certain data quality of the input, that is, the texts are expected to not be overly sparse and noisy to derive useful structured knowledge. The system also does not take time into consideration, so that a fixed, compact structured knowledge is derived without capturing the evolving patterns of knowledge. The Hierarchy Generation Algorithm requires a minimum human intervention, i.e., a user-specified root tag concept. The algorithm also uses a greedy, layer-by-layer, approach to form the hierarchies. The siblings, or concepts having the same parent, were not consistently modelled in the hierarchy. This may be further addressed through a globally optimised process to generate hierarchies. Some presented relations in the hierarchies were "related", but not strict subsumption relations. Due to the noisiness of social tagging data, while the discovered new knowledge can be used to enrich KBs, it needs scrutiny of domain experts. The evaluation process, especially the ontology-level evaluation, also assumes the existence of gold-standard structured knowledge, which may not be the case in all the knowledge domains.

Despite the conditions and limitations described above, the proposed machine learning system can be potentially adapted to other types of social media data. It is worth to apply and adapt the Data Representation and Feature Generation modules to other socially shared texts, such as microblogs, comments, and questions created by users in various types of social media platforms.

With the recent arise of deep learning for language processing, one of the future works is to apply deep learning models to improve the quality of the discovered knowledge. Another future work is to adapt the current supervised learning method to an online learning framework in order to build evolving structured knowledge. In this way, the learned hierarchy can update itself with the availability of new tagging data taking into consideration temporal factors. The design would also help capture the emerging semantics in a timely manner.

One key purpose of learning high quality structured knowledge is to support downstream semantic-based applications. In the next chapter, we will present a deep learning model that *leverages* structured knowledge for automated social annotation, which also addresses the incompleteness issue of users' tagging in many social media platforms.

Chapter 4

Knowledge-Enhanced Deep Learning for Social Annotation

...a birder sees a "robin" when a normal person only sees a "bird". – Paul Heymann and Hector Garcia-Molina [80], based on the work by James W. Tanaka and Marjorie Taylor [173]

One reason why titles and prefaces are ignored by many readers is that they do not think it important to classify the book they are reading. They do not follow this first rule of analytical reading. If they tried to follow it, they would be grateful to the author for helping them. Obviously, the author thinks it is important for the reader to know the kind of book he is being given. – Mortimer J. Adler and Charles Van Doren [2, p. 63]

Knowledge plays a key role in many semantic-based, machine learning applications by providing contextual information, enhancing explainability and improving performance. Among the many applications, automated social annotation can alleviate the incompleteness issue of social tagging data and help maintain data quality. In this chapter, we focus on leveraging structured knowledge to support the task of automated social annotation. The task is recently and more commonly formulated as a multi-label classification problem and modelled using deep learning approaches. We first introduce the task in 4.1 and then provide a problem formulation of the task as multi-label classification in 4.2.

The main challenge then is to leverage structured knowledge in deep learning models for multi-label classification. Multi-label classification needs to take into consideration the label correlation, i.e. relation among labels. For automated social annotation, the relation among labels is a type of structured knowledge of the user-generated tags. In Section 4.3, semantic-based loss regularisation is proposed to enhance the deep learning model with the similarity and subsumption relations between tags. Besides, to mimic the users' reading and annotation behaviour, a new form of attention mechanisms, guided

attention mechanisms, is further introduced to learn to guide the reading of sentences through the title metadata. The overall proposed deep learning model, Joint Multi-label Attention Networks (JMAN), can leverage the relations between tags, and separately models the title and the content of each document and injects an explicit, title-guided attention mechanism into each sentence. The approach has been evaluated with four real-world datasets from paper annotation and question annotation in social media platforms. Experiments are presented in Section 4.4 with analysis on model convergence, parameter tuning, multi-source components, and attention visualisation. The related work, mostly regarding the deep learning methods for automated social annotation and the attention mechanisms, is then reviewed in 4.5 with comparison to the proposed approach. The summary and discussion are in Section 4.6.

4.1 Introduction

The idea of automated social annotation was briefly introduced in Section 2.4.1 as an important semantic-based application. We recap the concept here with further introduction. As stated earlier, user-generated tags, or folksonomies, are collaboratively contributed by many users in social media platforms, beneficial for retrieval and recommendation of resources [184]. While tags are originally created by users, it is natural to consider, with a collection of documents and their associated tags, whether it is possible to automatically annotate new documents. The task of *automated social annotation* thus aims at predicting a set of tags based on the input metadata of a document shared in a social media platform. Figure 4.1 displays an example of a published paper and its associated user-generated tags on Bibsonomy.

The task can tackle the incompleteness and improve the overall quality of social tagging data. In reality, social tagging data face a serious issue of data incompleteness, as reviewed in Section 2.2.1. A substantial amount of socially shared documents online are not annotated with any (hash-)tags, e.g. around 20% of question in Zhihu [130] and at least 85% of microblogs (or tweets) on Twitter [97, 191]. Automatic annotation can support users' tagging process, reduce their cognitive load in tagging, enrich tag sets for the resources and result in a more stable quality of resource organisation in social media platforms [11, 90, 130]. These together constitute the motivations of automatic annotation of social texts. The task is relevant to indexing information resources, i.e. allocating of terms to describe resources after content analyses [134, p. 120]. The task can also be considered as a type of *object-centred tag recommendation*, which aims at enhancing the quality of tagging and benefit information retrieval in social media platforms in general. An example of the social annotation task is the recent data science competition, "Zhihu Machine Learning Challenge 2017"[1], to automatically annotate questions in Zhihu, one of the leading social question and answering sites in China.

[1] https://biendata.com/competition/zhihu/

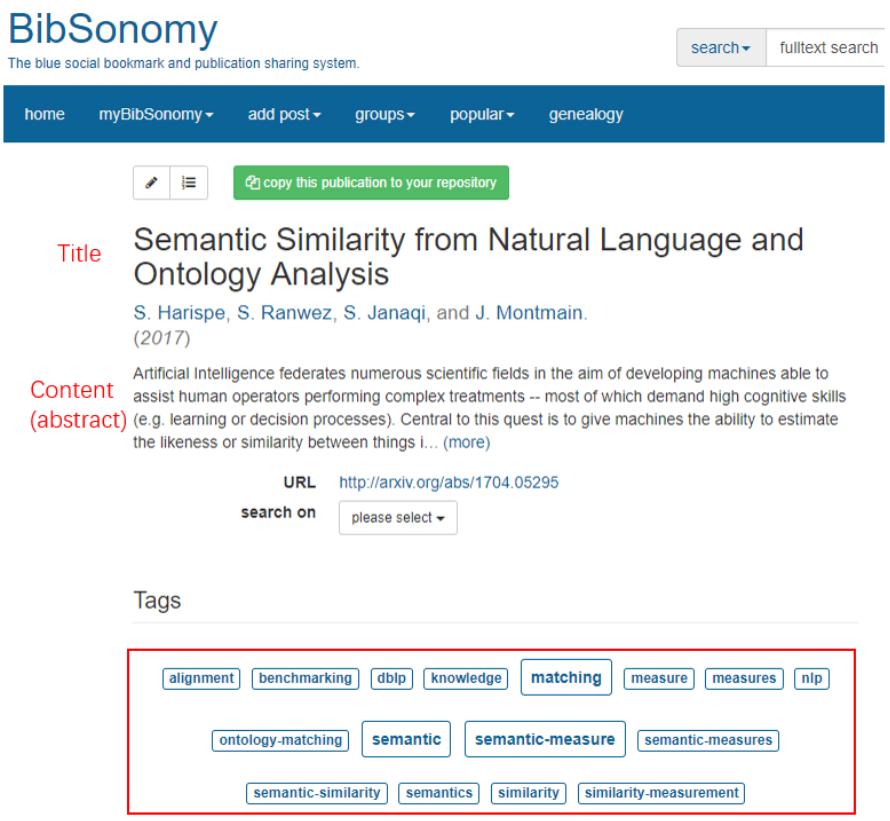


FIGURE 4.1: An example of a document and its associated metadata and tags on Bibsonomy. The metadata consist of the title and the content (i.e. the abstract of the paper). Tags are surrounded with a red box.

The structured knowledge of tags provides contextual information to improve the performance of automated social text annotation. In the "Zhihu Machine Learning Challenge 2017", the hierarchy of tags, as a DAG, were provided as additional structured knowledge to be leverage to enhance the annotation. However, as far as we concern, none of the winning teams in the competition leveraged the structured knowledge in their modelling process, instead, most teams used ensemble learning over many deep learning architectures to match the input document space to the label space[2] [162]. The lack of knowledge-enhanced approach in the competition calls for further exploration in this area. This corresponds to the original research question on how to leverage structured knowledge for semantic-based applications, especially for automated social annotation.

To accurately annotate documents with user-generated tags, another key challenge is to model the users' reading and annotation process. The user-generated tags may not appear in the document (e.g. papers or questions) and the number of tags is large. In the Bibsonomy dataset, after a thorough data cleaning, there were still above 5,000 cleaned tag concepts. Also, a document can be associated with many tags, for example in the cleaned Bibsonomy dataset described in Section 3.6.1.1, the average number of tag concepts per document was about 12. Recent advances in deep learning have demonstrated its superior performance in multi-label classification and text classification

[2]There were 1068 teams participated (4185 participants), and 211 teams in the final leaderboard[3].

[104, 128, 208, 209]. Recent studies explored the use of deep learning based approaches with attention mechanisms [67, 77, 89, 110, 210], which encode the input texts as continuous vector representations and approximate the matching from the input to the label space, where labels are often assumed to be orthogonal or independent to each other. Attention mechanisms, initially applied in neural machine translation [9] to form distinct context vector with respect to the target word to decode, are able to select important words and sentences in a document to improve text classification. An example of using attention mechanisms is in Hierarchical Attention Network (HAN) [200]. We will further introduce deep learning architectures, especially architectures of Recurrent Neural Networks (RNN) [52], [68, Chapter 10] with attention mechanisms in the proposed method Section 4.3 and the related work Section 4.5.

The existing deep learning methods for this task, however, at least suffer two issues: the modelling of reading and annotation behaviour (encoding) and the semantics in the labels (label correlation):

- In prediction, the most common *multi-hot* (as opposed to one-hot) representation for each label set [128] (see Section 2.4.4, and Section 4.2 below) assumes orthogonality among labels and does not consider their correlation. Label correlation is, however, a key issue in multi-label classification especially when the label size is large [62, 209], as reviewed in Section 2.4.4. In automated social annotation, the label correlation is based on the semantic relations among tags; and the co-occurring tags often exhibit similarity or subsumption relations [133, 164]. It is necessary to incorporate the knowledge of tag relations to model the annotation process.

- In encoding, mainstream methods simply scan the texts in the document and do not fully model the way how users read and annotate it. Recurrent Neural Networks (RNN) typically encode a sequence of text one word by another into a fixed length vector, while not considering the internal structure of documents. Hierarchical Attention Network (HAN) [200] models the hierarchical (word-sentence) structure of a document, however, it does consider how a document is annotated by a human user with the presence of different metadata, e.g. a user may digest the title before reading the document. Studies have explored the impact and importance of title on users' annotation choice [114], document categorisation and tag recommendation [54].

Further to add regarding the first issue, in the social context, users tend to annotate documents collectively with tags of various semantic forms and granularities [80, 133]. Current studies mostly considered the symmetric relation (similarity) among labels (tags) [65, 102, 213]. The asymmetric relation (such as subsumption) among labels needs further exploration, as suggested in [213]. To incorporate both types of label semantics in one deep neural network, we propose two semantic-based loss regularisers

in Section 4.3.1, along with the binary cross-entropy loss, to constrain the network output to satisfy the similarity and subsumption relations among labels. The regularisers allow the model to leverage semantic relations matched to existing KBs and inferred from datasets. We further explore the dynamic update of the semantic relations when optimising the loss regularisers.

We finally present a novel knowledge-enhanced and attention-based deep learning framework in Section 4.3 to seamlessly integrate users' reading and annotation behaviour in the encoding and prediction for automated annotation, leveraging the guided attention mechanisms and the label correlation encoded in external knowledge sources. We propose a new form of attention mechanisms to simulate users' reading behaviour. To annotate a document, a user attempts to digest the meaning of the title first; then, based on her or his understanding, proceeds to the content (e.g. abstract of the document). The key is the use of a title-guided attention mechanism that allows the meaning of the title to govern the "reading" of each sentences to form a final representation of the document. The idea is different from the attention mechanism used in the HAN model which is implemented through an implicit vector. In our approach, the guided attention mechanism is realised through a dynamic alignment of the title and sentences, which also enables better explainability in the modelling and visualisation.

4.2 Problem Statement: Multi-Label Classification

The task of automated social annotation can be formally transformed into a *multi-label classification* problem [128, 209], where each instance is associated with a set of labels instead of a single label in *multi-class classification*. In the scenario of social annotation, an object is most likely annotated with several user-generated tags instead of one single tag, thus multi-label classification is a suitable formulation for this task.

Suppose X denoting the collection of textual sequences or instances (e.g. documents), and $Y = \{y_1, y_2, ..., y_n\}$ denotes the label space with n possible labels (i.e. user-generated tags). Each instance in X, $x \in \mathbb{R}^{d_e}$, is a word sequence, in which each word is represented as a d-dimensional vector. Each x is associated with a label set $Y_i \subseteq Y$. Each $\overrightarrow{Y_i}$ is an n-dimensional *multi-hot* vector, $\overrightarrow{Y_i} = [y_{i1}, y_{i2}, ..., y_{in}]$ and $y_{ij} \in \{0, 1\}$, where a value of 1 indicates that the jth label y_j has been used to annotate (is relevant to) the ith instance, whereas a value of 0 indicates irrelevance of the label to the instance [128]. The task is to learn a complex function $h : X \rightarrow Y$ based on a training set $D = \{x_i, \overrightarrow{Y_i} | i \in [1, m]\}$, where m is the number of instances in the training set [209].

4.3 The Proposed Approach

Following the problem formulation above, we propose a deep learning model, a parallelled, two-layered attention network (Joint Multi-label Attention Network, JMAN) to model the users' reading and annotation process. The JMAN model is illustrated in

Figure 4.2 below. Instead of feeding the whole text sequence X into the neural network as in Hierarchical Attention Network (HAN) [200], the proposed model JMAN takes as inputs the title, x_t, and the content, x_a, separately, where $x = \{x_t, x_a\}$. Each target is a multi-hot vector, $\overrightarrow{Y_i} \in \{0,1\}^{|Y|}$. There are four attention mechanism modules, shown as dotted edges in Figure 4.2: two word-level attention mechanisms for the words in the title and in each sentence in the content, respectively; and two sentence-level attention mechanisms, one guided by the title representation ("title-guided") and the other guided by an "informative" vector ("original"). JMAN's key distinctions from the previous models are:

- The *semantic-based loss regularisers* aim to enhance the learning process by enforcing the output of the network to conform to the label correlation, i.e. leveraging the structured knowledge represented as similarity and subsumption relations, as specified in KBs (Section 4.3.1).
- The architecture of *multi-source hierarchical attention mechanisms* adapts the Hierarchical Attention Network (HAN) [200] to allow multiple input sources to specify different metadata or textual features of a document in different ways in parallel (Section 4.3.2).
- The *guided attention mechanisms*, specifically, title-guided, sentence-level attention mechanisms, that explicitly model the reading behaviour of users during annotation (Section 4.3.3).

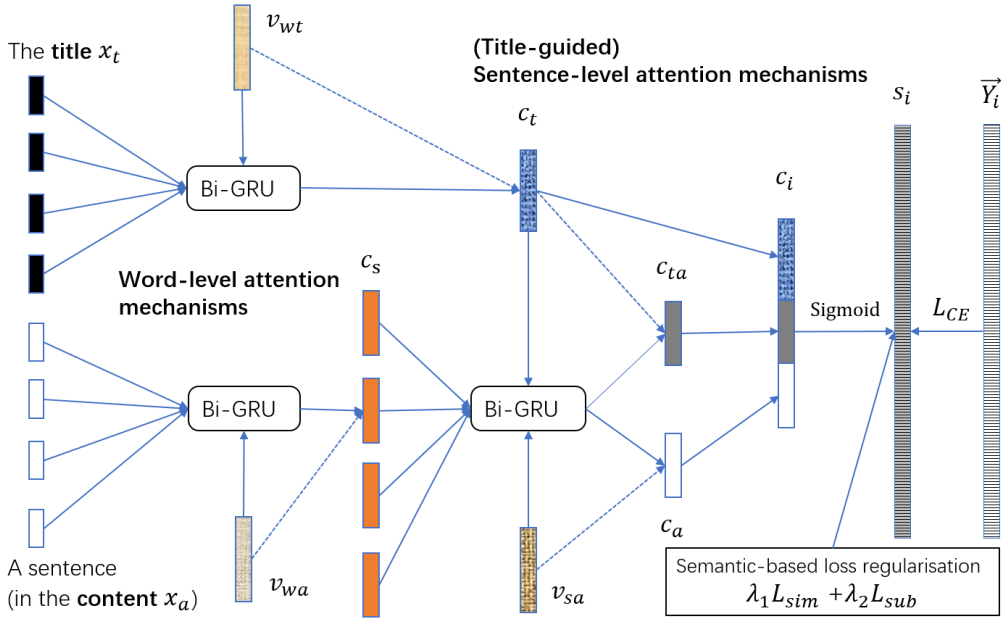


FIGURE 4.2: The proposed Joint Multi-label Attention Network (JMAN) for automated social annotation

4.3.1 Semantic-based Loss Regularisers

Studies show that tags have hidden semantic structures (e.g similarity and subsumption) and users collectively annotate documents with semantically related tags of various forms and granularities [80, 90, 133, 164]. If we treat each tag (here tag means tag concepts) as a label, then we have to take the label correlation into account for multi-label classification. Leveraging the label correlation is particularly challenging as the number of relation pairs might be enormously large when there are many labels [209]. In this case, it is computationally inefficient (if not infeasible) to apply the weight initialisation approach [10, 102] (as we introduced in Section 2.4.4) that assigns a neuron in the penultimate layer of the neural network to "memorise" just one of the numerous label relations.

We take a different strategy by using the semantic-based loss regularisation to leverage the structured knowledge of the labels. Two loss regularisers are proposed to deal with the similarity and subsumption relations, respectively, jointly optimised with the binary cross-entropy loss. The idea is to enforce the output values of the neural network $s_i \in (0,1)^{|Y|}$, of the same dimensionality as the label space, to satisfy the semantic constraints from the label relations. Such relations can be inferred from the label sets in the data and through grounding the labels to external KBs. The whole joint loss is defined as in Equation 4.1 below:

$$L = L_{CE} + \lambda_1 L_{sim} + \lambda_2 L_{sub} \tag{4.1}$$

where L_{CE} is the *binary cross entropy* loss [128], which obtained superior results with faster convergence over the *pairwise ranking* loss proposed in [208] for multi-label text classification with a feed-forward neural network. In Equation 4.2 below, $y_{ij} \in \{0,1\}$ indicates the true value whether a label $y_j \in Y$ has been used to annotate the ith document, and s_{ij} is the actual output value after the sigmoid layer.

$$L_{CE} = -\sum_i \sum_j (y_{ij} \log(s_{ij}) + (1 - y_{ij}) \log(1 - s_{ij})) \tag{4.2}$$

While the binary cross-entropy loss defines the matching between the output values and the target label set, the proposed L_{sim} and L_{sub} shown in Equation 4.3 define how the output values internally conform to the label relations defined in external KBs and induced from the dataset.

$$\begin{aligned} L_{sim} &= \frac{1}{2} \sum_i \sum_{j,k|y_j,y_k \in Y_i} Sim_{jk} |s_{ij} - s_{ik}|^2 \\ L_{sub} &= \frac{1}{2} \sum_i \sum_{j,k|y_j,y_k \in Y_i} Sub_{jk} R(s_{ij})(1 - R(s_{ik})) \end{aligned} \tag{4.3}$$

where Y_i is the set of all labels for the ith document; j and k are the indices of a co-occurring pair of labels, y_j and y_k, in the label set Y_i, corresponding to the indices of nodes s_{ij} and s_{ik} in the output layer s_i in Figure 4.2. $R()$ represents the rounding function for binary prediction, $R(s_{ij}) = 0$ if $s_{ij} < 0.5$, otherwise $R(s_{ij}) = 1$.

The label similarity matrix, $Sim \in (0,1)^{|Y|*|Y|}$, stores pairwise label similarity, the larger the value of Sim_{jk}, the more similar the labels y_j and y_k are to each other. Each element Sub_{jk} in the label subsumption matrix, $Sub \in \{0,1\}^{|Y|*|Y|}$, indicates whether the label y_j is a child label of y_k. Both the Sim and Sub matrix can be inferred from the data and from external KBs before the training. In the implementation, Sim (if a threshold is used for all entries) and Sub can be treated as sparse matrices in matrix multiplication to reduce computational complexity.

The idea for L_{sim} is that, in collective tagging, besides the same labels, users tend to collectively annotate documents with different labels that have very similar meanings. In multi-label learning, labels with high semantic similarity tend to be predicted together with similar values. The L_{sim} is a multiplication between two terms, Sim_{jk} and $|s_{ij} - s_{ik}|^2$. To minimise L_{sim}, intuitively, for very similar co-occurring labels y_j and y_k, i.e. with high Sim_{jk} close to 1, their corresponding nodes in the output layer should have minimal difference so that $|s_{ij} - s_{ik}|^2$ is low; for labels having low similarity with Sim_{jk} close to 0, there is almost no strict requirement on their corresponding nodes in the output layer, since, in multiplication, the squared difference $|s_j - s_k|^2$ will be scaled down with a low similarity value. Thus, the lower the similarity of the labels, the less enforcement on their corresponding nodes is required in the output layer to minimise the loss. L_{sim} has a similar but distinct form to the label manifold regulariser in [213]: the latter considers minimising differences of vector representations for low-rank approximation, while L_{sim} minimises node differences in the output layer of a neural network.

The idea for L_{sub} is that, in collective tagging, besides the same labels, users often annotate documents using different labels with different levels of specificity based on their knowledge and understanding. An analogy for this "basic level difference" across individuals [173] is that "a birder sees a 'robin' when a normal person only sees a 'bird' " [80, p. 4]. In the case of social annotation, for example, a researcher from the machine learning area would share and annotate using "LSTM", but researchers from other areas may annotate the same paper using more general labels such as "Neural Networks" or "Machine Learning". Distinct from similarity relations, the subsumption relations between labels are asymmetric. For two tags having a subsumption relation, if the child tag is associated with the document, there is a relatively higher likelihood that the parent tag is also related to the same document. In L_{sub}, if two labels having a subsumption relation $< y_j \rightarrow y_k >$ are both present in the label set Y_i, the case that the parent label y_k is predicted as false (i.e. $R(s_{ik}) = 0$), when its child label y_j is predicted as true (i.e. $R(s_{ij}) = 1$), will be penalised. Such a case will result in a positive penalty, while the penalty will be 0 in all other cases.

Thus, along with the well-established binary cross-entropy loss, the L_{sim} enforces semantically similar labels to have similar output values, while L_{sub} reinforces each co-occurring subsumption pair to according to the dependency of the parent label on the child label. We finally optimise the joint loss function in Equation 4.1 with the L_2 regularisation using the Adam optimiser [99].

In practice, one potential adaptation of the semantic-based loss regularisers is to dynamically update the Sim and Sub matrices, as the pre-defined relations between labels may not be compatible with the semantics of the labels in the dataset. In doing this, both Sim and Sub become continuous representations and can have negative entries. This adds a further, "negative" constraint to the last layer s_i (see Figure 4.2) of the neural network. Taking L_{sim} as the example: the more negative the value of Sim_{jk}, the less similar the labels y_j and y_k, then according to Equation 4.3, the case of $|s_{ij} - s_{ik}|^2$ being large (i.e., label y_j and y_k having very different predicted probability) will be favoured. A similar constraint is added with L_{Sub}: the more negative the value of Sub_{jk}, the less strong the subsumption relation is from y_j to y_k, then the case that $R(s_{ij})(1 - R(s_{ik}))$ being 1 (i.e., label y_j predicted as true and label y_k predicted as false) will be favoured. Dynamic updating of Sim and Sub, however, requires further substantial memory, especially for a large number of labels. We first focus on the fixed Sim and Sub and compare the results between dynamic and fixed Sim and Sub in the main experiments.

This novel joint loss function can be broadly used in many deep learning architectures to support multi-label classification. We will describe the multi-source hierarchical attention mechanisms, along with the state-of-the-art architectures, Bi-GRU and HAN, adapted to this joint multi-label learning approach in the following section.

4.3.2 Multi-Source Hierarchical Attention Mechanisms

To better capture the different types of metadata that users' read for annotation, we model the title and the content separately, distinct to the original HAN architecture [200] and the recent work on socially paper annotation [77], as titles are the key textual features which greatly influence the choice of tagging [114] and the performance of classification and annotation [54]. This multi-source hierarchical attention architecture constitutes the backbone of the JMAN model, as described below.

4.3.2.1 Embedding Layers

Each input title or content (usually multiple sentences) is an ordered set of words, represented as $x_t = (v_t^{(1)}, v_t^{(2)}, ..., v_t^{(\mathrm{n_t})})$ and $x_a = (v_a^{(1)}, v_a^{(2)}, ..., v_a^{(\mathrm{n_a})})$, respectively, where $\mathrm{n_t}$ or $\mathrm{n_a}$ denotes the number of words in the title or content, respectively. The embedding layer transforms the input vocabularies v into low-dimensional vectors, which are formally defined as $e_t = W_e v_t$, $e_a = W_e v_a$, where $W_e \in \mathbb{R}^{d_e \times |V|}$ is the embedding weights that are usually pre-trained via neural word embedding algorithms, e.g., Word2Vec [125] or Glove [132]. The embedding dimensionality is far less than the vocabulary size $|V|$,

i.e. $d_e \ll |V|$. The more recent contextualised word embeddings, ELMo [136], can also be applied to this layer.

4.3.2.2 Bi-GRU Layers

We adapt Recurrent Neural Networks (RNN) to encode the documents and establish the matching to the labels. The theoretical foundation is that RNN is also a universal approximator as multi-layer feed-forward neural networks [85], with a mathematical prove provided in [149]. A problem in the vanilla RNN is the vanishing gradient, e.g., when reading a lengthy sequence, the RNN "reader" may forget the previous words before it completes processing the whole sequence. Long Short-Term Memory (LSTM) [83] and Gated Recurrent Units (GRUs) [35] have been proposed to address this problem. GRUs have been applied to the original HAN model [200] and to neural machine translation [9], which are efficient in training and can achieve a similar level of performance to LSTM. We follow this setting and use GRUs as the basic recurrent unit. The Bi-GRU layer processes text sequences in both directions.

GRUs introduce two gates, a reset gate $r^{(t)}$ and an update gate $z^{(t)}$, to control and generate a new hidden state $h^{(t)}$ from the previous hidden state $h^{(t-1)}$. RNN with GRUs can be formally defined in Equations 4.4, where σ refers to a non-linear activation function (here we use the logistic sigmoid function), and $W_{er}, W_{ez}, W_{e\tilde{h}} \in \mathbb{R}^{d_h \times d_e}$, $W_{hr}, W_{hz}, W_{h\tilde{h}} \in \mathbb{R}^{d_h \times d_h}$ are weights, where d_h is the number of hidden units. We use the GRU model with bias terms $b_r, b_z \in \mathbb{R}^{d_h}$ as in [200], shown in the Equations 4.4 below, where the $\circ$ denotes the Hadamard product or the element-wise product.

$$
\begin{aligned}
r^{(t)} &= \sigma(W_{er}e^{(t)} + W_{hr}h^{(t-1)} + b_r) \\
z^{(t)} &= \sigma(W_{ez}e^{(t)} + W_{hz}h^{(t-1)} + b_z) \\
\tilde{h}^{(t)} &= \tanh(W_{e\tilde{h}}e^{(t)} + W_{h\tilde{h}}(r^{(t)} \circ h^{(t-1)})) \\
h^{(t)} &= (1 - z^{(t)}) \circ h^{(t-1)} + z^{(t)} \circ \tilde{h}^{(t)}
\end{aligned}
\tag{4.4}
$$

The idea of Bidirectional-RNN [151] with GRUs [35], denoted as Bi-GRU, is proposed to capture the fact that a word in a sequence is not only related to its previous words, but also to its following words. Bi-GRU consists of forward GRUs and backward GRUs. The forward GRUs read the embedding of each word in the input sequentially from left to right, e.g. from $e^{(1)}$ to $e^{(\mathrm{n})}$, to produce forward hidden states ($\overrightarrow{h^{(1)}}$,...,$\overrightarrow{h^{(\mathrm{n})}}$); whereas the backward GRUs read the sequence reversely from $e^{(\mathrm{n})}$ to $e^{(1)}$ to calculate backward hidden states ($\overleftarrow{h^{(\mathrm{n})}}$,...,$\overleftarrow{h^{(1)}}$). Both hidden states are concatenated to construct a new fixed-length vector as the output hidden state, $h^{(i)} = [\overrightarrow{h^{(i)}}; \overleftarrow{h^{(i)}}]$. In the proposed network (see Figure 4.2), after the reading in both directions is completed, the title and content are represented as context vectors $\mathbf{c}_t$ or $\mathbf{c}_a$, respectively. These vectors are normally set as the last concatenated hidden states $h^{(n)}$; however, doing so tends to emphasise the

words towards the end of the sequence. Therefore, the attention mechanisms [9, 200] need to be applied to re-calculate the vectors $\mathbf{c}_t$ or $\mathbf{c}_a$.

4.3.2.3 Hierarchical Attention Layers

Attention mechanisms have been widely used in natural language processing tasks, since the study [9] on machine translation in 2014. Instead of encoding a long sequence (such as sentences and paragraphs) into a single vector representation for all times, they allow the neural networks to learn to focus on part of the input sentence each time aligned with a different context, e.g. the next target word in the translation [9]. More recently, attention mechanisms have been applied to encode documents for classification, taking into consideration the hierarchical structure of documents, as proposed in the HAN model [200].

The idea of Hierarchical Attention is closely related to how users read and comprehend documents. The HAN model [200] generally assumes that, to understand a document, users read the document word by word in each sentence, then sentence by sentence. During reading and annotation, users would pay special attention to the most informative words or sentences, which might be considered to annotate that document later. There are three Bi-GRU layers in JMAN as shown in Figure 4.2, each accompanied by its attention layer(s): two word-level attention layers, for title and sentences in the content, respectively; and two sentence-level attention layers, one is *original* sentence-level attention layer proposed in [200] and the other is the *title-guided* sentence-level attention mechanism (see Section 4.3.3).

To model the different amount of attention a user paid on each word or sentence, a weighted average of hidden representations is applied as suggested in [9, 200]. The attention scores are based on an alignment of each hidden representation in a sequence of words or sentences to a non-static and learnable, "informative" vector representation, which is supposed to encode "what is the informative word or sentence" in the sequence [200] and commonly used in document classification tasks [77, 101]. We apply the dot product as the alignment measure [117, 200]. The word-level attention mechanism models the different importance of each word in the title or in a sentence, while the sentence-level attention mechanism makes a distinction for each of the sentences. The word-level attention mechanism in the title (and similarly in sentences) is described in Equations 4.5 below.

$$
\begin{aligned}
v^{(i)} &= \tanh(W_t h^{(i)} + b_t) \\
\alpha^{(i)} &= \frac{\exp(v_{wt} \bullet v^{(i)})}{\sum_{i \in [1, \mathrm{n_t}]} \exp(v_{wt} \bullet v^{(i)})} \\
c_a &= \sum_{i \in [1, \mathrm{n_t}]} \alpha^{(i)} h^{(i)}
\end{aligned}
\tag{4.5}
$$

where a fully connected layer is added to transform the hidden state $h^{(i)}$ to a vector representation $v^{(i)}$, followed by an alignment to the attention vector v_{wt} with the dot product operation (denoted as $\bullet$). A softmax function is then applied to obtain the attention weights $\alpha^{(i)}$. The context vector c_a, which is the representation of the sequence, is computed as the weighted average of all hidden state vectors $h^{(i)}$. In a similar way, we can compute the word-level attention mechanism for each sentences as well as the original sentence-level attention mechanism (for details refer to the HAN model in [200]). The attention vectors v_{wt}, as well as v_{wa} and v_{sa} in Figure 4.2, represent "what is the informative word or sentence" during the reading process [200]. They are randomly initialised and jointly learned during the training.

4.3.3 Guided Attention Mechanisms on the Sentence Level

The attention mechanisms above are not considered enough to make a clear distinction among sentences. Firstly, the impact of the title metadata on the document annotation is not modelled, which is however particularly important during the user tagging process [54, 114]. Secondly, in the attention mechanisms described in Equation (4.5) above, the "informative" vector v_{wt}, commonly treated as weights to be learned in the model, as in other recent studies [77, 101], does not reflect any explicit object in humans' reading and understanding.

We find that the implicit "informative" attention weight vectors, such as v_{wt}, can be made clearer through conjecturing the reading order. Selection of the important sentences in the content should ideally conform to the main theme of the document, of which, a reasonable source is the title. Title is a short, abstractive summarisation and a good starting point to understand a document. This title-guided sentence-level attention mechanism, as shown in Figure 4.2, can be modelled as in Equations (4.6):

$$\begin{aligned} v_s^{(r)} &= \tanh(W_s h_s^{(r)} + b_s) \\ \alpha_s^{(r)} &= \frac{\exp(c_t \bullet v_s^{(r)})}{\sum_{k\in[1,\mathrm{n_s}]} \exp(c_t \bullet v_s^{(k)})} \\ c_{ta} &= \sum_{r\in[1,\mathrm{n_s}]} \alpha_s^{(r)} h_s^{(r)} \end{aligned} \tag{4.6}$$

where $h_s^{(r)}$ is the hidden state of the rth sentence; c_t is the title representation obtained from Equation (4.5); $\mathrm{n_s}$ denotes the total number of sentences in the content; $\alpha_s^{(r)}$ is the sentence-level attention score; W_s, b_s are learnable weights in the network. This title-guided attention mechanism is distinct from the recent, concurrent work in [34], which uses title information on the word level to enhance the annotation for keyphrase generation. The "title-guided encoding" in the study [34] calculates, for each word in the document, a different title representation, to be later concatenated with the word;

the approach has shown to improve the performance on keyphrase generation, but is not based on the assumption on human's reading and annotation in this chapter.

Guiding the sentences solely with the title may cause the final content representation to be overly dependent on the title. The actually content of a document usually contains (far) more detailed information not described in the title, which can help suggest more related tags during annotation [54]. For example, some sentences in the content (or abstract) can highlight an innovative and important evaluation study which may not be presented in the title. To avoid an overemphasis on the effect of the title for annotation, we concatenate the representation generated from the title-guided sentence-level attention mechanism with the original one; thus we can form a more comprehensive content representation. The final document representation is the concatenation of the title representation with the content representation, $c_i = [c_t, c_{ta}, c_a]$, as illustrated in Figure 4.2. We will show the effectiveness of this design, in terms of both performance and convergence speed, by comparing against its baselines and variations.

4.4 Experiments

In this section, we describe our experiments using four real-world datasets from two types of social annotation applications: paper annotation in academic social tagging systems, Bibsonomy and CiteULike (two variations), and question annotation in a social question & answering site, Zhihu. Performance comparison shows the significant performance gain of JMAN over the current state-of-the-art models in terms of the evaluation metrics, with a substantial improvement of convergence speed. We will also discuss the impact of the regularisation parameters and analyse the attention mechanisms through visualisation. The code for the work, implementation details, and all the cleaned datasets are openly available at GitHub[4].

4.4.1 Datasets

We chose the benchmark social tagging datasets in the academic domain, Bibsonomy and CiteUlike, and the social Q&A site in the general domain, Zhihu, for our experiments. On Bibsonomy and CiteUlike, users can share publications and annotate them with tags. Metadata of the documents such as title and abstract (or content) are also available. We directly used the cleaned Bibsonomy dataset, preprocessed with the Data Cleaning module described in Section 3.2 and Section 3.6.1.1, and then selected the documents containing both the title and the abstract. For better qualitative analysis, we further selected the documents having at least one tag matched to the concepts in the ACM Computing Classification System[5].

[4] `https://github.com/acadTags/Automated-Social-Annotation`
[5] `https://www.acm.org/publications/class-2012`

For CiteUlike, we used the benchmark datasets, CiteULike-a and CiteULike-t, released in [187]. We applied similar preprocessing steps as the Bibsonomy dataset[6] and further removed the tags occurring less than 10 times. If a tag has insufficient usage (low frequency), then it is difficult to learn to annotate it to other documents due to scarcity in the data. This ensures enough training data for each tag and significantly reduced the dimensionality of the label space.

Zhihu is a leading Chinese social Q&A site in all domains. Tags are used on Zhihu to describe the topics of questions, and to support searching and recommending questions and answers for users. Each question has a title and/or a detailed description (or content). We downloaded the official benchmark open data from the Zhihu Machine Learning Challenge 2017[7], containing more than 3 million questions, together annotated with 1,999 unique labels. The Zhihu dataset had been preprocessed before its release: all the Chinese words had been segmented and replaced with an unknown codebook due to privacy issues. We randomly sampled around 100,000 questions having both the title and the content; this ensured a sufficient amount of data with a reasonable training time per fold, compared to the other three datasets (see Table 4.4).

To extract the subsumption relations for all tags in each of the datasets (except Zhihu), we grounded the tags to concepts and instances in the external KB, Microsoft Concept Graph (MCG)[8]. MCG is a data-driven KB which has around 1.8M concepts and instances, and 8.5M subsumption relations. Zhihu released its crowdsourced tag hierarchies which can be directly used to as subsumption relations between labels. It is also possible to ground the tags to other KBs, such as DBpedia, or use the structured knowledge induced from the tagging data. We did not directly use the learned knowledge from Chapter 3 due to (i) the relative smaller size of learned subsumption relations compared to those grounded to MCG, (ii) the learned knowledge still needs scrutiny from domain experts, considering the precision, recall and F_1 score and analysis through hierarchy visualisation. This warrant further exploration on exploiting various knowledge sources and end-to-end approaches to jointly learn and leverage structured knowledge.

Statistics of the cleaned datasets are shown in Table 4.1, including number of documents $|X|$, number of labels $|Y|$, vocabulary size in documents $|V|$, average number of labels per document Ave and the number of label subsumption pairs for each dataset Σ_{Sub}. The average number of labels per document in the social Q&A dataset (Zhihu) is much less than the paper annotation datasets (Bibsonomy and CiteULike), but the former has a larger number of documents and vocabulary size. The number of labels in all datasets is large, from around 2.0K to around 5.2K. The number of subsumption relations grounded to MCG is also large, all above 100K except Zhihu. There are over 2.5K crowdsourced subsumption relations in Zhihu.

[6]Processing script implemented in Python for the CiteULike datasets is available on `https://github.com/acadTags/Tag-Data-Cleaning`.

[7]`https://biendata.com/competition/zhihu/`

[8]`https://concept.research.microsoft.com/Home`

TABLE 4.1: Multi-label datasets for social annotation

Dataset	$\|X\|$	$\|Y\|$	$\|V\|$	Ave	Σ_{Sub}
Bibsonomy (clean)	12,101	5,196	17,619	11.59	101,084
CiteULike-a (clean)	13,319	3,201	17,489	11.60	107,273
CiteULike-t (clean)	24,042	3,528	23,408	7.68	141,093
Zhihu (sample)	108,168	1,999	62,519	2.45	2,655

4.4.2 Experiment Settings

To calculate the label similarity matrix Sim for the semantic-based loss regularisers in Equations 4.3, we used cosine similarity (further normalised to between 0 and 1) of the pre-trained skip-gram embeddings [125] on all label sets in each dataset. To construct the label subsumption matrix Sub, we used the obtained label subsumption pairs from MCG and Zhihu. The values of λ_1 and λ_2 in L were tuned based on results from 10-fold cross-validation.

We implemented the proposed JMAN model and several popular and state-of-the-art baseline approaches on Tensorflow [1] and other Python packages. Seven baselines, including some downgraded models of JMAN, were chosen,

1. SVM-ovr: an one-versus-rest multi-label Support Vector Machine with word embedding features, implemented using the scikit-learn Python package[9]. We used the RBF kernel and tuned the C and γ according to [88] to achieve the best F_1 score on the validation sets. We used the same word embeddings as for JMAN, then the input is a document embedding as the average of word embeddings in the padded input document. We chose SVM as it usually performed better than many other classifiers used in document classification [39]. This baseline was also used in [110].

2. LDA: the probabilistic topic modelling approach, Latent Dirichlet Allocation (LDA) [19], was applied to represent each document as a probability distribution over hidden topics, implemented with the wrapper in the Python Gensim package [145] for the JAVA-based MALLET toolkit [121]. The algorithm was adapted to multi-label classification by assigning each new document the tags of its k most similar documents based on the document-topic distributions p(topic|document). We trained the LDA model for 1,000 iterations and tuned the number of topics T as 200 and k as 1 for all datasets based on the performance on the validation sets. The baseline was also used in [157].

3. Bi-GRU: the Bidirectional-RNN [151] with Gated Recurrent Units (GRUs) [35] for multi-label classification. The algorithm treats the title and the content together as the input sequence. The document representation $\mathbf{c_i}$ is set as the last concatenated hidden state.

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html`

4. HAN: the Hierarchical Attention Network in [200] and also applied in [77] for tag recommendation. We combined the title and the content, and fed into the HAN model as in [77]. This is a state-of-the-art model for document classification.

5. JMAN-s: the proposed model without semantic-based loss regularisers.

6. JMAN-s-tg: the proposed model without semantic-based loss regularisers and the title-guided sentence-level attention, i.e. $c_i = [c_t, c_a]$.

7. JMAN-s-att: the proposed model without semantic-based loss regularisers and the original sentence-level attention, i.e. $c_i = [c_t, c_{ta}]$.

8. JMAN$_d$: the proposed model with the dynamic update of Sim and Sub during training.

We trained all the models using 10-fold cross-validation and then tested on a separate, fixed 10% randomly held-out dataset. The number of hidden units, learning rate, and dropout rate [161] were set as 100, 0.01 and 0.5, respectively, for all models. The batch size for the Bibsonomy and CiteULike-a/t dataset was set to 128, and the batch size for the Zhihu dataset was set to 1,024. The sequence lengths of the title (also the length of each sentence) and the content were padded to 30 and 300 for Bibsonomy, CiteULike-a, and CiteULike/t; 25 and 100 for Zhihu. We parsed the sentences of Bibsonomy and CiteULike-a/t based on punctuation and padded the sentences to a fixed length. For Zhihu, as the data had been masked, we simply set a fixed length to split the content into "sentences". Non-static input embedding for the title and the sentences were initialised as a 100-dimension pre-trained skip-gram embedding [125] from the documents. We decayed the learning rate by half when the loss on the validation set increased and set an early stopping point when the learning rate was below a threshold (2e-5 for Bibsonomy and Zhihu; 1e-3 for CiteULike-a and CiteULike-t). Experiments on all neural network models were run on a GPU server, NVIDIA GeForce GTX 1080 Ti (11G GPU RAM), except for the dynamic update of Sim and Sub on Intel® Xeon® Processor E5-2630 v3 or v4 with 60G RAM; experiments on SVM-ovr and LDA were run on an Intel® Xeon® CPU E5-1620 v2 3.70GHz.

We also implemented three problem transformation algorithms, i.e. transforming the features or label space for multi-label classification with a base classifier, Classification Chain (CC) [141, 142], Hierarchy Of Multilabel classifiER (HOMER) [177], and Principal Label Space Transformation (PLST) [170], adapting the Python scikit-multilearn [169] wrapper of MEKA [143] (based on WEKA [75] and MULAN [179]). The base classifier was chosen as SVM with RBF kernel for the methods. Due to large numbers of documents and labels considered, the program took much longer than the SVM-ovr implementation in Table 4.4 and required substantial memory. With the default parameters in MEKA, the results of the three methods were not better than the results of the SVM-ovr classifier. We thus do not report their results in this chapter, but provide an open implementation for reproducibility.

4.4.3 Evaluation Metrics

Five widely used example-based metrics for multi-label classification were applied to evaluate the models, including Hamming loss, Accuracy, Precision, Recall, F-measure, to assess the performance of the algorithms [65, 158, 178, 209]. For the metrics below, D_t denotes instances in the testing data, $f(x_i)$ and y_i denote the predicted and actual labels for the ith instance respectively.

- Hamming loss (H) measures the number of misclassified labels. It is defined as $\mathrm{H}(f) = \frac{1}{|D_t|}\sum_{i \in D_t} \frac{1}{Q}|f(x_i)\Delta y_i|$, where Δ is the symmetric difference between two sets and Q is a normalisation constant. We set Q as the average number of labels per document, Ave, in the data (see Table 4.1). The lower the value, the better the performance.

- Accuracy (A), defined as the fraction of the correctly predicted labels to the total number of labels presented (union of predicted and actual ones), computed as $\mathrm{A}(f) = \frac{1}{|D_t|}\sum_{i \in D_t} \frac{|f(x_i) \cap y_i|}{|f(x_i) \cup y_i|}$.

- Precision (P), defined as the fraction of the correctly predicted labels to all the predicted labels, $\mathrm{P}(f) = \frac{1}{|D_t|}\sum_{i \in D_t} \frac{|f(x_i) \cap y_i|}{|f(x_i)|}$.

- Recall (R), defined as the fraction of the correctly predicted labels to all the actual labels, $\mathrm{R}(f) = \frac{1}{|D_t|}\sum_{i \in D_t} \frac{|f(x_i) \cap y_i|}{|y_i|}$.

- F-measure (F_1), defined as the harmonic mean between precision and recall, $F_1(f) = \frac{2P(f)R(f)}{P(f)+R(f)}$.

4.4.4 Evaluation and Comparison

We presented the evaluation results using the metrics and compared the performance of JMAN to the popular, state-of-the-art classification models, and the downgraded variants of the model. In particular, we highlighted the performance of using the semantic-based loss regularisers.

4.4.4.1 Main Results

Table 4.2 shows the evaluation and comparison results using JMAN and the other baseline approaches on the four datasets[10]. The proposed model JMAN and JMAN$_d$ performed the best in terms of accuracy and F_1 score, and among the top or comparably well in terms of precision, recall, and Hamming Loss, on all datasets. Most results of JMAN$_d$ were better than JMAN on the Bibsonomy and CiteULike-a/t datasets, together showing a further absolute increase of F_1 from 0.2% to 1.2% over JMAN, indicating the

[10]We were not able to obtain the results of SVM-ovr on the Zhihu dataset as the training time was extremely long. Training time per each fold (in 10-fold cross-validation) could take over 1 day in some parameter settings, which prevented efficient training and parameter tuning. JMAN$_d$ also required substantial memory and we failed to obtain results with the specified settings on the Zhihu datasets.

TABLE 4.2: Comparison results of JMAN and others on the four social annotation datasets in terms of Hamming Loss(H), Accuracy(A), Precision(P), Recall(R), and F_1 score (F_1)

		SVM-ovr	LDA	Bi-GRU	HAN	JMAN-s-tg	JMAN-s-att	JMAN-s	JMAN	JMAN$_d$
Bib	H	*107.7±0.2*(8)	*142.3±2.0*(9)	*90.1±0.7*(7)	*86.1±0.4*(6)	***84.5±0.5***(1)	*84.6±0.3*(2)	85.2±0.5(4)	85.1±0.6(3)	85.2±0.5(4)
	A	19.2±0.2(9)	*21.0±0.5*(7)	*19.2±1.3*(8)	*22.0±1.0*(6)	*24.1±0.6*(5)	*24.2±0.6*(4)	24.8±0.4(3)	25.1±0.4(2)	**25.4±0.5**(1)
	P	39.2±0.3(8)	*31.1±0.8*(9)	*52.2±2.0*(7)	*57.2±0.8*(6)	59.1±1.0(2)	**59.2±1.0**(1)	58.6±0.4(5)	58.8±0.8(3)	58.6±0.5(4)
	R	*25.2±0.2*(7)	***31.1±0.7***(1)	*21.7±1.6*(9)	*24.6±1.2*(8)	*26.9±0.6*(6)	*27.2±0.7*(5)	*28.2±0.5*(4)	28.6±0.3(3)	28.9±0.6(2)
	F_1	*30.7±0.2*(8)	*31.1±0.7*(7)	*30.6±1.9*(9)	*34.4±1.3*(6)	*37.0±0.7*(5)	*37.3±0.8*(4)	*38.0±0.5*(3)	38.5±0.4(2)	**38.7±0.5**(1)
C-a	H	*118.1±0.3*(8)	*168.2±1.5*(9)	*100.0±0.7*(7)	96.0±0.5(5)	*94.6±0.5*(2)	***94.5±0.3***(1)	95.5±0.5(3)	95.7±0.6(4)	*97.2±1.3(6)*
	A	*8.6±0.1*(8)	*9.5±0.3*(7)	*7.5±1.6*(9)	*11.0±0.8*(6)	13.5±0.6(4)	13.4±0.4(5)	13.6±0.8(3)	13.9±0.8(2)	**14.4±0.6(1)**
	P	*26.1±0.2*(8)	*18.5±0.5*(9)	*32.6±4.5*(7)	*42.9±1.4*(6)	47.9±1.2(2)	***48.4±0.8***(1)	47.2±1.6(4)	47.3±1.5(3)	47.1±1.1(5)
	R	*12.3±0.1*(8)	***18.6±0.6***(1)	*8.9±2.0*(9)	*13.2±1.1*(7)	*16.3±0.8*(5)	*16.0±0.6*(6)	16.6±1.2(4)	17.0±1.1(3)	*17.8±0.7(2)*
	F_1	*16.7±0.1*(8)	*18.6±0.5*(7)	*14.0±2.9*(9)	*20.2±1.4*(6)	24.3±1.0(4)	*24.1±0.7*(5)	24.6±1.5(3)	25.0±1.3(2)	**25.8±0.8(1)**
C-t	H	*113.5±0.3*(8)	*171.8±2.2*(9)	*97.1±0.7*(7)	***93.6±0.3***(1)	*94.2±0.3*(3)	*94.0±0.4*(2)	95.2±0.5(4)	95.2±0.6(5)	96.3±0.8(6)
	A	*8.7±0.2*(9)	*9.2±0.2*(8)	*10.9±2.3*(7)	*11.9±1.0*(6)	*13.6±0.6*(4)	*13.5±0.3*(5)	14.4±0.6(3)	14.5±0.4(2)	***15.2±0.8(1)***
	P	*24.5±0.3*(8)	*17.2±0.2*(9)	*34.9±5.1*(7)	*38.2±1.8*(6)	*39.8±1.2*(5)	*40.0±0.8*(4)	40.9±0.9(2)	40.9±0.6(3)	***42.3±0.9(1)***
	R	*12.2±0.2*(9)	*17.7±0.5*(3)	*13.0±2.9*(8)	*13.8±1.3*(7)	*16.2±0.8*(5)	*16.2±0.4*(6)	17.6±0.9(4)	17.8±0.7(2)	***18.7±1.0(1)***
	F_1	*16.3±0.2*(9)	*17.4±0.3*(8)	*18.9±3.9*(7)	*20.3±1.7*(6)	*23.0±1.0*(4)	*23.0±0.5*(5)	24.6±1.0(3)	24.8±0.7(2)	***26.0±1.1(1)***
Zhi	H	-	*187.9±0.7*(7)	95.3±0.3(5)	***93.4±0.2***(1)	*94.3±0.3*(2)	*94.6±0.3*(3)	95.3±0.5(6)	95.2±0.6(4)	-
	A	-	*3.9±0.2*(7)	*13.9±0.8*(6)	15.3±0.8(4)	15.5±0.3(3)	15.3±0.4(5)	**15.6±0.5**(1)	**15.6±0.5**(1)	-
	P	-	*5.6±0.2*(7)	*23.8±1.1*(6)	25.7±1.2(2)	25.7±0.5(4)	25.4±0.7(5)	25.7±0.8(3)	**25.8±0.9**(1)	-
	R	-	*5.6±0.2*(7)	*15.4±0.9*(6)	*16.7±1.0*(5)	17.5±0.3(3)	*17.4±0.5*(4)	17.7±0.5(2)	**17.8±0.6**(1)	-
	F_1	-	*5.6±0.2*(7)	*18.7±1.0*(6)	20.3±1.1(5)	20.8±0.3(3)	*20.7±0.5*(4)	21.0±0.7(2)	**21.1±0.7**(1)	-

For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold**. The results in *italics* indicate that the difference between JMAN and others is statistically significant with paired t-tests at a 95% significance level. The number in round brackets "()" shows ranking of the algorithm.

usefulness of the dynamic update of the label semantic matrices *Sim* and *Sub*. The results of JMAN were significantly better (denoted in *italics*) than HAN and Bi-GRU in terms of accuracy, precision, recall and F_1 score, with few exceptions on the Zhihu dataset for HAN.

Comparing to other deep learning models, in terms of F_1, JMAN provided an absolute increase up to 11.0% (by 78.6%) and 4.8% (by 23.7%) over Bi-GRU, and HAN for the CiteULike-a dataset; and 5.9% (by 31.2%) and 4.5% (by 22.2%) over Bi-GRU and HAN for the CiteULike-t dataset. A similar performance gain was achieved using the Bibsonomy dataset, with an absolute increase of 7.9% (by 25.8%) over Bi-GRU and 4.1% over HAN (by 11.9%); and a relatively smaller increase using the Zhihu datasets of 2.4% (by 13.4%) over Bi-GRU, and 0.8% over (by 3.4%) HAN. This overall improvement showed that the separate modelling of the metadata, and the title-guided attention on the sentences, clearly boosted the performance on automated annotation. The results of HAN were better than Bi-GRU in most settings, which showed the effectiveness of modelling the hierarchical pattern of a document with attention mechanisms, and validated the results in [200].

JMAN also outperforms its several downgraded models. Effectiveness of the semantic-based loss regularisers was observed by comparing the results produced by JMAN and JMAN-s (without semantic-based loss regularisers). The regularisers helped improve the recall and F_1, although with a relatively low margin. The results of JMAN are significantly better than JMAN-s-tg and JMAN-s-att, where either the title-guided or the original sentence-level attention mechanism is removed, in terms of accuracy, precision and F_1 score in most evaluation settings.

Only little improvement was observed with the Zhihu dataset, largely due to its distinct characteristics compared to other datasets: Zhihu has much shorter texts (around 1/3 of the texts in other datasets), larger vocabularies (about 3-4 folds), fewer number of labels (around 40%-60%) and fewer average number of labels per document (around 20%-30%), as shown in Table 4.1. We also noticed that the result of Hamming Loss was not always consistent with the other four metrics. Hamming Loss measures the symmetric difference between two sets, which treats every label equally; while the example-based metrics, Accuracy, Precision, Recall and F_1 score, are scaled by the length of the actual label set and/or the predicted label set. From the results, we observed that the relative difference of Hamming loss among HAN, JMAN and its downgraded variants, JMAN-s, JMAN-s-tg and JMAN-s-att, were all marginal.

Comparing to the models SVM-ovr and LDA, the JMAN model and its variations performed significantly better in terms of all metrics for all the datasets, except a few cases where the LDA resulted in higher recall but lower precision and F_1 score. Although SVM-ovr and LDA can achieve a better F_1 score than Bi-GRU with high recall on the Bibsonomy and CiteULike-a datasets, they performed poorly in terms of Hamming Loss. The result of LDA for the Zhihu dataset was rather worse, which may be because the users' annotation process could not be well modelled through topic-based similarity

among questions, or the data statistics as stated above that made it difficult to learn better topic representations for the questions. The results, in overall, have demonstrated the significant improvement of JMAN over the popular and state-of-the-art baselines.

4.4.4.2 Results on Semantic-Based Loss Regularisers

To test the effectiveness of the semantic-based loss regularisers L_{sim} and L_{sub}, that leverage label relations by constraining the output of neural networks, we applied them (either separately or collectively) with the fixed *Sim* and *Sub* setting[11] on Bi-GRU, HAN and JMAN-s, and reported the results on the testing data using models trained with 10-fold cross-validation.

From Table 4.3, it can be seen that models with the semantic-based loss regularisers (either one or both), consistently performed better than the original models. 0.9% to 1.6% absolute gain of F_1 from the four datasets was observed for Bi-GRU, and 0.6% to 1.6% for HAN. For the JMAN-s model, the improvement with semantic-based loss regularisers is less obvious; there was only 0.1% to 0.5% absolute increase of F_1. It is hard to draw a clear conclusion on which of the L_{sim} and L_{sub} was more effective further improving the model performance in multi-label social text annotation. This may depend on the hidden label structure from the data, i.e., which of the semantic relations, similarity or subsumption, is more prominent in the label sets. From the results, we can see that L_{sim} and L_{sub} complement to each other and achieved the best results in half (6 out of 12) of the experimental settings, for the other cases, using either one L_{sim} or L_{sub} performed better than using them together. It was also noticed that results of Hamming Loss are not consistent with the other metrics and differences among the models' Hamming Losses are marginal, as in Table 4.2.

The results produced by adding the semantic-based loss regularisers indeed coincided with our initial perception and expectation that model performance could be further improved by leveraging structured knowledge as label correlation with the help of external KBs. However, most of the differences in the evaluation settings were not statistically significant. The evaluation result was generally in line with the one produced in the existing research that also leveraged label correlation in multi-label classification. The work using a weight initialisation approach in [10] reported performance gain of less than 1% in F_1 in most experimental settings. The proposed approach is more feasible than the weight initialisation approach [10] for data with large label sizes, typically in the context of automated social annotation, as explained in Section 2.4.4.

The marginal improvement from the experiments was probably due to the fact that the shared weights in the layers prior to the output layer in the neural networks may already and indirectly model some of the correlations among the output nodes, through solely approximating the matching from documents to labels. This might also explain

[11]The "dynamic" version was not fully tested due to the substantial memory required, although our preliminary experiments on the "dynamic" version show further improved results (see comparison between JMAN$_d$ and JMAN in Table 4.2).

TABLE 4.3: Comparison results of using the semantic-based loss regularisers on different deep learning models for the four social annotation datasets in terms of Hamming Loss(H), Accurary(A), Precision(P), Recall(R), and F_1 score (F_1)

		Bi-GRU	+L_{sim}	+L_{sub}	+both	HAN	+L_{sim}	+L_{sub}	+both	JMAN-s	+L_{sim}	+L_{sub}	+both(JMAN)
Bib	H	90.1±0.7	90.2±0.4	**89.7±0.6**	90.0±0.9	86.1±0.4	86.1±0.5	86.0±0.6	**85.9±0.5**	85.2±0.5	85.1±0.6	**84.6±0.7**	85.1±0.6
	A	19.2±1.3	19.5±0.7	19.5±0.7	**20.1±0.5**	22.0±1.0	22.2±0.7	22.5±0.5	**22.5±0.8**	24.8±0.4	24.9±0.5	**25.2±0.6**	25.1±0.4
	P	52.2±2.0	52.4±1.7	52.7±1.5	**53.3±1.7**	57.2±0.8	**57.3±1.2**	57.1±1.0	57.3±1.1	58.6±0.4	58.4±0.8	**59.2±0.9**	58.8±0.8
	R	21.7±1.6	22.1±0.9	21.9±0.9	**22.8±0.6**	24.6±1.2	24.7±0.8	25.2±0.7	**25.2±0.9**	28.2±0.5	28.4±0.5	28.5±0.7	**28.6±0.3**
	F_1	30.6±1.9	31.0±1.1	31.0±1.1	**31.9±0.8**	34.4±1.3	34.6±0.9	35.0±0.8	**35.0±1.1**	38.0±0.5	38.2±0.6	**38.5±0.8**	38.5±0.4
C-a	H	100.0±0.7	**99.2±0.8**	100.3±0.5	99.6±0.4	96.0±0.5	**95.5±0.4**	95.9±0.5	95.7±0.4	**95.5±0.5**	95.9±0.8	95.9±0.6	95.7±0.6
	A	7.5±1.6	**8.5±1.1**	7.7±1.2	8.2±1.3	11.0±0.8	11.4±0.8	11.0±0.6	**11.5±0.5**	13.6±0.8	13.8±0.7	13.8±0.6	**13.9±0.8**
	P	32.6±4.5	**35.8±3.3**	32.8±3.3	35.2±3.7	42.9±1.4	**43.8±1.2**	42.7±1.1	43.4±0.1	47.2±1.6	47.1±1.3	46.9±1.1	**47.3±1.5**
	R	8.9±2.0	**10.0±1.3**	9.2±1.5	9.7±1.6	13.2±1.1	13.6±1.0	13.2±0.8	**13.7±0.7**	16.6±1.2	**17.1±1.0**	17.0±0.9	17.0±1.1
	F_1	14.0±2.9	**15.6±1.9**	14.3±2.1	15.2±2.4	20.2±1.4	20.7±1.3	20.2±1.0	**20.9±0.9**	24.6±1.5	**25.1±1.2**	24.9±1.1	25.0±1.3
C-t	H	97.1±0.7	96.6±0.5	96.9±0.6	**96.4±0.3**	93.6±0.3	**93.5±0.2**	93.6±0.3	93.6±0.3	95.2±0.5	95.3±0.7	**95.1±0.5**	95.2±0.6
	A	10.9±2.3	**11.8±0.8**	11.0±1.2	11.8±0.4	11.9±1.0	12.4±0.6	**12.8±0.6**	12.4±1.0	14.4±0.6	**14.5±0.4**	14.4±0.5	**14.5±0.4**
	P	34.9±5.1	36.8±1.5	35.4±2.5	**37.4±1.2**	38.2±1.8	38.7±0.8	**39.4±0.9**	38.6±1.8	40.9±0.9	41.1±0.6	**41.1±0.8**	40.9±0.6
	R	13.0±2.9	**13.9±1.1**	13.0±1.5	13.9±0.7	13.8±1.3	14.5±0.8	**15.1±0.9**	14.5±1.4	17.6±0.9	17.7±0.8	17.7±0.8	**17.8±0.7**
	F_1	18.9±3.9	20.2±1.3	19.0±2.0	**20.3±0.9**	20.3±1.7	21.1±0.9	**21.9±1.1**	21.1±1.7	24.6±1.0	24.7±0.8	24.7±0.9	**24.8±0.7**
Zhi	H	**95.3±0.3**	95.4±0.4	95.5±0.4	95.4±0.3	93.4±0.2	**93.3±0.2**	**93.3±0.2**	93.4±0.3	95.3±0.5	**95.1±0.4**	95.3±0.3	95.2±0.6
	A	13.9±0.8	**14.6±0.3**	14.4±0.7	14.3±0.5	15.3±0.8	**15.7±0.5**	15.6±0.7	**15.7±0.5**	**15.6±0.5**	15.6±0.3	15.6±0.2	**15.6±0.5**
	P	23.8±1.1	**24.9±0.5**	24.7±1.0	24.5±0.8	25.7±1.2	**26.5±0.7**	26.3±1.1	26.4±0.9	25.7±0.8	**25.9±0.5**	25.8±0.5	25.8±0.9
	R	15.4±0.9	**16.2±0.4**	16.1±0.9	15.9±0.6	16.7±1.0	**17.3±0.6**	17.0±0.8	17.2±0.6	17.7±0.5	17.8±0.4	17.8±0.2	**17.8±0.6**
	F_1	18.7±1.0	**19.6±0.5**	19.5±1.0	19.3±0.7	20.3±1.1	**20.9±0.7**	20.7±0.9	20.8±0.7	21.0±0.7	21.1±0.4	21.1±0.3	**21.1±0.7**

* For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold** for each category of models separated by column lines. Bib, C-t, C-a, and Zhi denote the Bibsonomy, CiteULike-t, CiteULike-a, and Zhihu datasets, respectively.

** The *Sim* and *Sub* matrices in L_{Sim} and L_{Sub} were fixed during training.

why JMAN-s is less boosted by the regularisers than Bi-GRU and HAN: JMAN-s better models the document encoding part, and could thus, compared to Bi-GRU and HAN, indirectly alleviate the label correlation issue to a greater extent. We also noticed that the work in [128] reported somehow different results, i.e. that the binary cross-entropy loss, L_{CE}, achieved better performance than the pairwise ranking loss [208] which implicitly considers label correlation. We believe that leveraging structured knowledge as label correlation for a wide array of multi-label classification tasks is necessary; while the semantic-based loss regularisers provide a useful approach to incorporate such knowledge, the problem remains challenging and needs further studies. As a potential approach towards this direction, we also found that the dynamic update of the label semantic matrices Sim and Sub (as in JMAN$_d$) can further improve the fixed setting of label semantics (as in JMAN), see the two rightmost columns of the Table 4.2, but at the cost of substantial memory. This dynamic update adds further constraints to the nodes in the output layer (with negative values in Sim and Sub) and allows the label semantics to be more compatible to the knowledge embedded in the dataset, as discussed in Section 4.3.1. This provides insights to leverage the dynamic structured knowledge of the labels and warrants further studies.

4.4.5 Training Time and Model Convergence

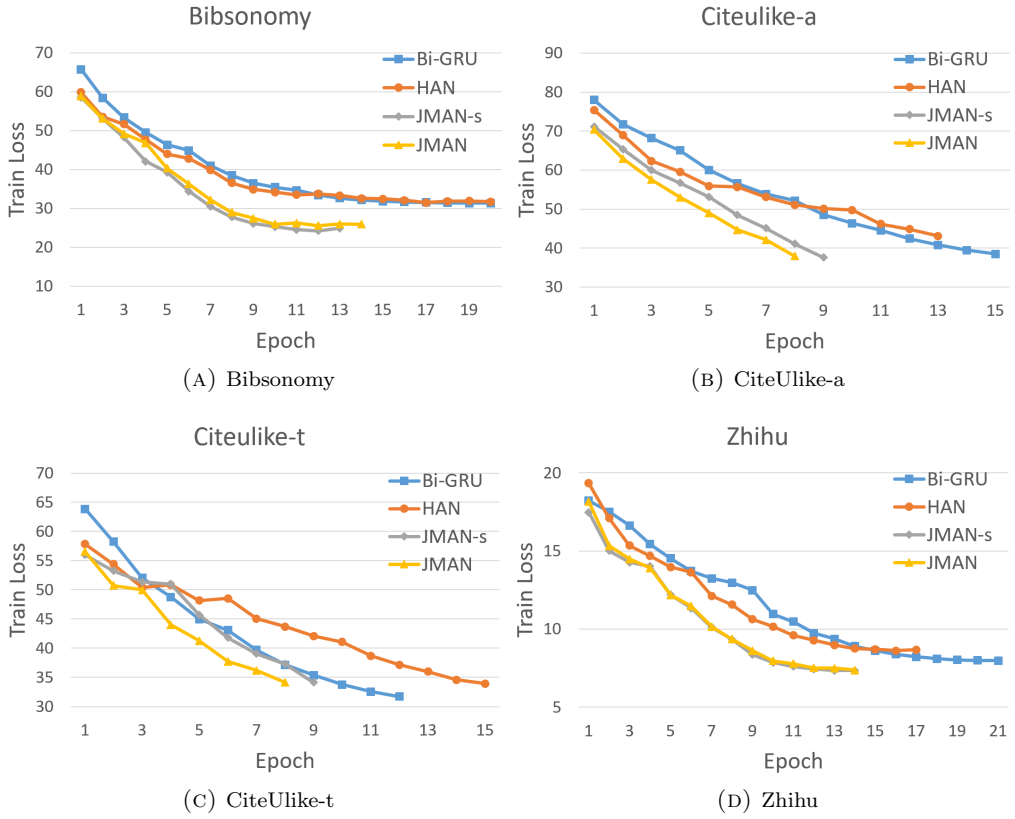
Table 4.4: Comparison of training time for the multi-label classification models in seconds

	Bib	C-a	C-t	Zhi
SVM	1107 ± 12	1660 ± 31	4796 ± 50	over 1 day
LDA	**110 ± 2**(1)	**113 ± 3**(1)	**210 ± 7**(1)	**903 ± 31**(1)
Bi-GRU	1480 ± 92	869 ± 288	1635 ± 1034	1455 ± 69
Bi-GRU+s	1683 ± 78	877 ± 57	1469 ± 276	2459 ± 151
HAN	1164 ± 52	462 ± 63	858 ± 100	1387 ± 78
HAN+s	1434 ± 74	554 ± 45	947 ± 115	2388 ± 275
JMAN-s-tg	1075 ± 87	434 ± 49	**752 ± 52**(3)	**1220 ± 81**(3)
JMAN-s-att	**1024 ± 100**(3)	**429 ± 41**(3)	780 ± 69	1275 ± 99
JMAN-s	**894 ± 55**(2)	**394 ± 33**(2)	**744 ± 62**(2)	**1147 ± 44**(2)
JMAN	1138 ± 86	468 ± 38	839 ± 49	1712 ± 105

Training time of the three most efficient models are in **bold** and marked with a ranking index in brackets "()". BiGRU+s and HAN+s denote the models with semantic-based loss regularisers.

In Table 4.4, we reported the mean and standard deviation of training time spent per fold for each model in the 10-fold cross-validation. With the efficient and highly scalable implementation of Gibbs sampling for approximation in MALLET [121], the LDA model took the least time for training. Among the other models, JMAN-s was the most efficient in training despite its relatively more complex architecture, by around 21.2%-54.7% faster than Bi-GRU and around 13.3%-23.2% faster than HAN on all datasets.

The training time increased when the semantic-based loss regularisers were used; the increased time was related to the document size $|X|$, label size $|Y|$ and the average length of the label sets *Ave* of the dataset (see Equations (4.3) and Table 4.1). The SVM-ovr model was the least efficient as it trained one SVM RBF classifier for every single label and the number of unique labels was large.



(A) Bibsonomy (B) CiteUlike-a

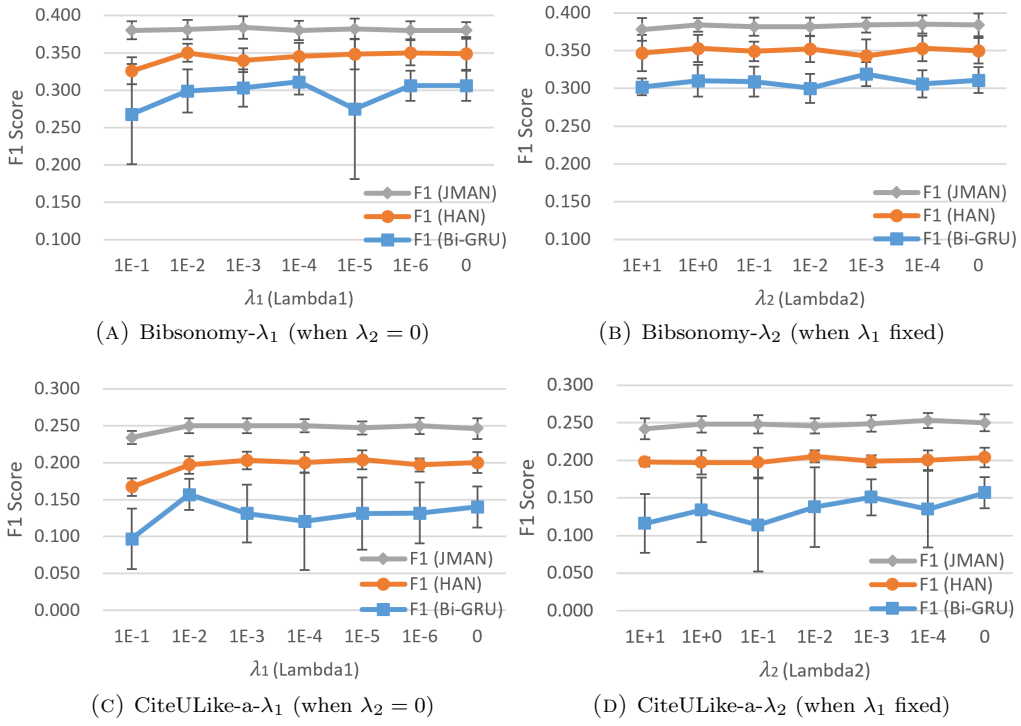
(C) CiteUlike-t (D) Zhihu

FIGURE 4.3: Convergence plot: training loss with respect to the number of training epochs for the Bi-GRU, HAN, JMAN-s, and JMAN models

The difference in training time among the neural network based models, Bi-GRU, HAN, JMAN-s, and JMAN, can also be explained by the convergence plots in Figure 4.3. The total number of epochs for each model was determined by early stopping based on a validation set. It can be observed that on all four datasets, JMAN and JMAN-s converged much faster than Bi-GRU and HAN, with fewer training epochs and steeper convergence plots. HAN also converges faster than Bi-GRU. It should be noted that the lower training loss does not necessarily imply better performance on the testing data (see testing results in Table 4.2). The convergence plots show that JMAN and JMAN-s can "understand" (or learn to represent) the input documents within fewer epochs than HAN and further than Bi-GRU.

4.4.6 Parameter Sensitivity Analysis

In the joint loss function in Equation 4.1, there are two regularisation parameters, λ_1 and λ_2, controlling the influence of the similarity and subsumption loss regularisers



(A) Bibsonomy-λ_1 (when $\lambda_2 = 0$)

(B) Bibsonomy-λ_2 (when λ_1 fixed)

(C) CiteULike-a-λ_1 (when $\lambda_2 = 0$)

(D) CiteULike-a-λ_2 (when λ_1 fixed)

FIGURE 4.4: F_1 score with respect to the λ_1 and λ_2 on Bibsonomy and CiteULike-a datasets using the Bi-GRU, HAN, and JMAN models

on training. A larger λ_1 means to exert more constraint on the output layer of the neural network to enforce similarity relations among labels, and a larger λ_2 means more constraints to enforce subsumption relations. We selected empirically reasonable sets of values for λ_1 and λ_2, and took a two-step parameter tuning process: first, varying $\lambda_1 \in \{1\text{E-}1, 1\text{E-}2, ..., 1\text{E-}6\}$ and setting λ_2 as 0 to find the best λ_1, and second, varying $\lambda_2 \in \{1\text{E+}1, 1\text{E+}0, ..., 1\text{E-}4\}$ while fixing the tuned λ_1.

We reported here F_1 scores with the changing values for λ_1 and λ_2 on the Bibsonomy and the CiteULike-a datasets with the three models, Bi-GRU, HAN, and JMAN. Similar patterns of parameter sensitivity were observed for the CiteULike-t and Zhihu datasets.

As showed in Figure 4.4, Bi-GRU was the most susceptible model to the semantic-based loss regularisers with different weights, indicated by large fluctuations of the F_1 score and standard deviation. In contrast, JMAN's performance was not affected much by the changing parameter values, and HAN's was affected moderately. This was also in accordance with the analysis in Section 4.4.4.2 that the performance of Bi-GRU and HAN was improved to a larger extent than JMAN.

When λ_1 and λ_2 are both set to 0 (see Figure 4.4), the models are equivalent to the original ones without using the semantic-based loss regularisers. When given large values to λ_1 (e.g. 1E-1) and λ_2 (e.g. 1E+1) the models overly emphasised on the label relations instead of document-label matching, and their performance degraded. While the performance of the models could be boosted with proper settings of the parameter values, the improvement was not significant and the parameter tuning process was not

trivial, as analysed in Section 4.4.4.2. This warrants further studies on leveraging the knowledge of label correlation in deep learning to enhance model performance in multi-label classification.

4.4.7 Analysis of Multi-Source Components

The architecture described in Section 4.3.2 combines the title representation c_t, content c_a, and title-guided content c_{ta}. It is worth analysing how different source of the representations contributes to the performance of annotation. Table 4.5 presents the results with c_t, c_a, c_{ta}, and different combinations of them on the four datasets, without the use of semantic regularisers. The JMAN-s model concatenates all three representations, while JMAN-s-tg and JMAN-s-att are combinations of title representation and one of the content representations. It is clear that the JMAN-s model, with the full representation of $[c_t, c_a, c_{ta}]$, performed the best among all models. A similar level of performance was observed in using JMAN-s-tg and JMAN-s-att, where either the title-guided content representation ("-tg") or the original content representation ("-att") was excluded. When only one type of the representation was used, the title-guided content representation performed the best. Although the title representation alone performed the worst, it boosts the annotation performance through guiding the representation of the content. While a single user may tend to provide annotations based on the title or the abstract only and browse the content selectively, their collective annotations tend to reflect the whole document. The results confirmed the advantage of using multi-source information for document representation.

4.4.8 Attention Visualisation

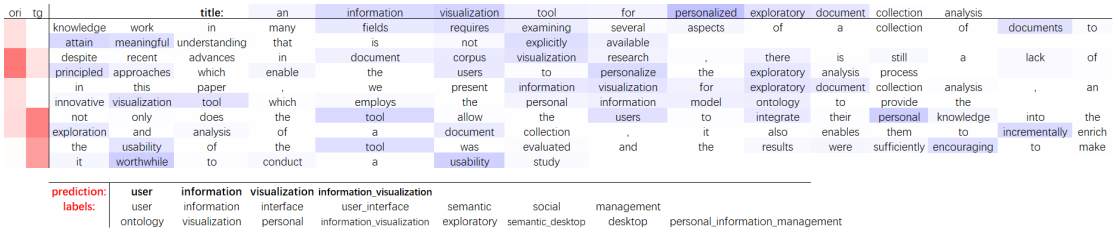
We can further understand how the hierarchical attention mechanisms work, especially the proposed guided attention mechanism, by visualising the attention weights (in Figure 4.5 on the next page). Four attention weights in JMAN were illustrated for sample documents from Bibsonomy, CiteULike-a and CiteULike-t: (1) word-level attention weights for title, i.e. the α in Equations 4.5, (2) word-level attention for each sentences in the abstract, (3) original sentence-level attention for the abstract, and finally (4) title-guided sentence-level attention weights for the abstract, i.e. the α_s in Equations 4.6. Documents and labels in the Zhihu dataset were not interpretable as all words had been officially masked with an unknown codebook.

In Figure 4.5, every two rows under the title is a sentence in the content (abstract). The red blocks in the two leftmost columns denote the sentence-level attention weights, where the left one ("ori") displays the *original* sentence-level attention weights and the right one ("tg") displays the *title-guided* sentence-level attention weights. The purple blocks denote the attention weights of each word in the title (the first row) or a sentence. The darker the colour, the greater the amount of attention was paid to a word or sentence in the model for annotation. The predicted labels by the JMAN model and the ground truth labels are shown below each diagram.

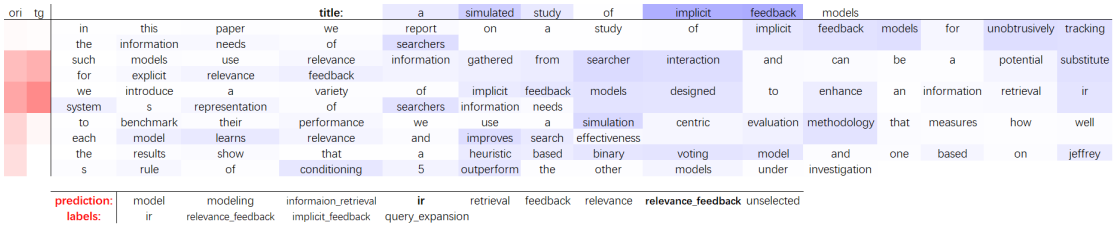
TABLE 4.5: Comparison results of multiple sources (title, content, and title-guided content representations) in the JMAN model on the four social annotation datasets in terms of Hamming Loss(H), Accuracy(A), Precision(P), Recall(R) and F_1 score (F_1)

		Title (c_t)	Content (c_a)	Content, title-guided (c_{ta})	JMAN-s-tg ($[c_t, c_a]$)	JMAN-s-att ($[c_t, c_{ta}]$)	JMAN-s ($[c_t, c_{ta}, c_a]$)
Bib	H	88.7 ± 0.8	87.7 ± 0.7	86.8 ± 0.5	**84.5 ± 0.5**	84.6 ± 0.3	85.2 ± 0.5
	A	17.0 ± 1.1	20.4 ± 1.1	21.2 ± 0.5	24.1 ± 0.6	24.2 ± 0.6	**24.8 ± 0.4**
	P	50.4 ± 1.6	54.7 ± 1.7	55.4 ± 0.6	59.1 ± 1.0	**59.2 ± 1.0**	58.6 ± 0.4
	R	18.4 ± 1.2	22.8 ± 1.3	23.7 ± 0.6	26.9 ± 0.6	27.2 ± 0.7	**28.2 ± 0.5**
	F_1	26.9 ± 1.5	32.2 ± 1.6	33.2 ± 0.7	37.0 ± 0.7	37.3 ± 0.8	**38.0 ± 0.5**
C-a	H	96.4 ± 0.2	97.1 ± 0.3	97.0 ± 0.3	94.6 ± 0.5	**94.5 ± 0.3**	95.5 ± 0.5
	A	7.3 ± 0.4	9.5 ± 0.5	9.6 ± 0.9	13.5 ± 0.6	13.4 ± 0.4	**13.6 ± 0.8**
	P	34.0 ± 1.5	39.2 ± 1.4	39.5 ± 1.4	47.9 ± 1.2	**48.4 ± 0.8**	47.2 ± 1.6
	R	8.3 ± 0.6	11.4 ± 0.7	11.5 ± 1.3	16.3 ± 0.8	16.0 ± 0.6	**16.6 ± 1.2**
	F_1	13.3 ± 0.8	17.6 ± 1.0	17.8 ± 1.7	24.3 ± 1.0	24.1 ± 0.7	**24.6 ± 1.5**
C-t	H	96.1 ± 0.3	95.4 ± 0.5	95.2 ± 0.3	94.2 ± 0.3	**94.0 ± 0.4**	95.2 ± 0.5
	A	5.7 ± 0.9	10.3 ± 0.4	10.5 ± 1.1	13.6 ± 0.6	13.5 ± 0.3	**14.4 ± 0.6**
	P	21.2 ± 2.5	33.3 ± 1.0	34.0 ± 1.7	39.8 ± 1.2	40.0 ± 0.8	**40.9 ± 0.9**
	R	6.5 ± 1.1	12.1 ± 0.6	12.3 ± 1.5	16.2 ± 0.8	16.2 ± 0.4	**17.6 ± 0.9**
	F_1	9.9 ± 1.5	17.8 ± 0.8	18.0 ± 1.9	23.0 ± 1.0	23.0 ± 0.5	**24.6 ± 1.0**
Zhi	H	97.0 ± 0.2	97.2 ± 0.2	94.9 ± 0.2	**94.3 ± 0.3**	94.6 ± 0.3	95.3 ± 0.5
	A	7.1 ± 0.8	7.4 ± 0.4	9.7 ± 0.8	15.5 ± 0.3	15.3 ± 0.4	**15.6 ± 0.5**
	P	12.2 ± 1.1	12.6 ± 0.7	17.2 ± 1.2	25.7 ± 0.5	25.4 ± 0.7	**25.7 ± 0.8**
	R	7.8 ± 0.9	8.1 ± 0.5	10.4 ± 0.9	17.5 ± 0.3	17.4 ± 0.5	**17.7 ± 0.5**
	F_1	9.5 ± 1.0	9.9 ± 0.6	13.0 ± 1.0	20.8 ± 0.3	20.7 ± 0.5	**21.0 ± 0.7**

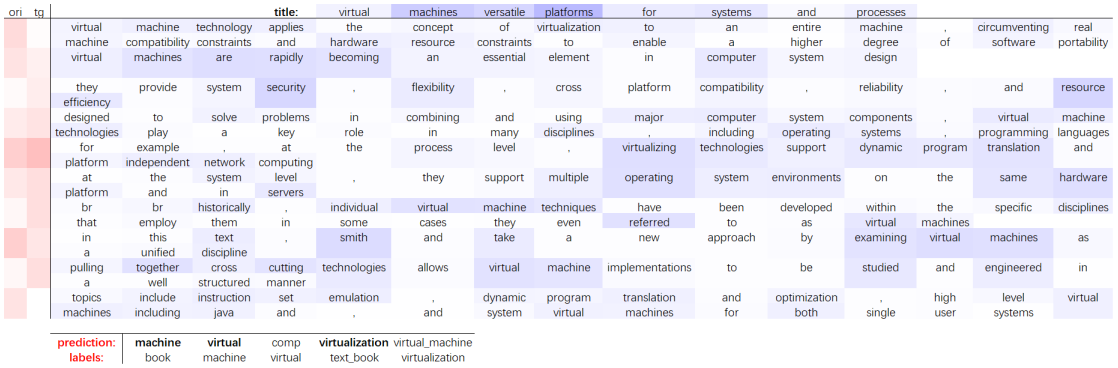
For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold**.



(A) Bibsonomy Example



(B) CiteULike-a Example



(C) CiteULike-t Example

FIGURE 4.5: Attention visualisation of the proposed JMAN model for the testing documents from the Bibsonomy, CiteULike-a, and CiteULike-t datasets. Red blocks in the leftmost two columns show the *original* ("ori") and the *title-guided* ("tg") sentence-level attention weights, respectively. Purple blocks mark the word-level attention weights for the title (the first row) and each sentence (every two rows) in the abstract. The darker the colour, the greater amount of attention was paid to the word or the sentence in JMAN. The predicted labels and the actual "ground truth" labels are displayed below each diagram.

It can be observed that the title-guided sentence-level attention ("tg") assigned different weights and provided a distinct "view" from the original sentence-level attention ("ori"). In the Bibsonomy example, the "ori" weights highlighted mostly the second sentence (a general statement that identifies the gap in the literature), while the "tg" weights highlighted more the fourth (a statement of a tool that allows integrating personal knowledge into exploration of a document collection) and fifth sentences (continuation of the previous statement on the tool's usability). These two sentences are well aligned to the title and intuitive for users to determine the main themes of the document for annotation. This difference was also present in the other two examples. As discussed in Section 4.3.3, concatenating the output from both attention mechanisms would help gain a more comprehensive understanding of the documents and provide more accurate

annotation (as indicated by the comparison results with JMAN-s-tg, JMAN-s-att, and JMAN-s in Table 4.2). This is because that the abstract of a document may contain more useful and important information that is not present in the title. For example, in the CiteULike-a example, the "tg" weights highlighted only the second and third sentences which aligned well to the title; while the "ori" weights also emphasised the fourth and fifth sentences which talked about the "simulation", "evaluation" and two specific models. Although they were not well aligned to the title, they represented important information for document understanding. There was also certain degree of agreement between the two attention weights, for instance, in the CiteULike-a example, both attention weights were low for the first sentence (a general introduction) and high for the second (more detail about the topic) and the third sentences (more on the authors' work). The degree of agreement was even higher in the CiteULike-t example.

Besides, the word-level attention indeed highlighted many of the most informative words (from either the title or sentences). These informative words were either the same as or highly related to the true labels or the topics of the document, for example, "information", "user", "personalised" and "visualisation" in the Bibsonomy example; "implicit", "feedback", "ir", "models", and "searcher" in the CiteULike-a example; and "machine", "virtualising", "platform", "virtual", and "operating" in the CiteULike-t example. Words that conveyed no meanings regarding the topics of the document, such as the stop words and many uninformative ones, such as articles ("the", "a"), be verbs ("is", "are"), prepositions ("in") and conjunctions ("and", "or", "for", "to"), were assigned nearly zero weight (e.g. white colour in the blocks).

We also noticed some potential limitations of visualising such attention weights, regarding their explainability and stability. While the attention weights seem to provide insights on selecting the important parts of the data and have been applied in many previous studies [9, 200], it is suggested in the recent study [91] that the weights are not easily interpretable as "explanations" for RNN-based text classification models. We also observe that the sentence-level attention weights are not stable among different runs of the algorithm. This would warrant further studies on the interpretation and analysis of the attention visualisation results.

Last but not least, from the predicted results, we can see that the JMAN model suggested meaningful labels[12]. The predicted labels had a substantial overlap with the "ground truth" labels (cleaned user-generated tags), but still have the potential for improvement, especially in terms of recall. We also noticed that the true labels also contained some that were useless or not related to the topics of the document, for example, "book" and "text_book" (expressing the type of the document) in the CiteULike-t example, which are probably difficult to be predicted solely from the title and the abstract. It was also very interesting to see that the predicted labels not included in the "ground truth" were indeed highly relevant to the themes of the documents, which should have been used for annotation, e.g. "information_retrieval", "retrieval",

[12]More prediction results on the testing examples are available on `https://github.com/acadTags/Automated-Social-Annotation`.

"modelling" and "relevance" in the CiteULike-a example, and "virtual_machine" in the CiteULike-t example. This shows that the proposed approach also has the potential to enhance the quality of existing annotations.

4.5 Related Work

In Section 2.4 previously, we organised some related work regarding the role of structured knowledge for automated social annotation. In this section, we further relate the proposed approach in this chapter to the literature, focusing on deep learning approaches for multi-label classification, and then review the recent, relevant studies on attention mechanisms.

Automated social annotation can be viewed as objected-oriented tag recommendation [11], which suggests tags to annotate objects (e.g. documents) to enhance the downstream information retrieval services in general. Previous studies on automated social annotation applied various methods and techniques for the tag recommendation task and modelling users' tagging process. A survey of approaches on tag recommendation was in [11], including *tag co-occurrence-based*, *content-based*, *matrix factorisation based*, *clustering-based*, *graph-based*, *learning to rank based* methods. Studies also have modelled the users' tagging process on social Q&A sites and microblogging services through term frequency based lexical features [203], adaptive hypergraph learning [130] and probabilistic graphical models [46, 187].

Recent studies mostly apply deep learning for automated social annotation and commonly formulate the task as a multi-label classification problem. The advantage of multi-label deep learning models lies in their relatively straightforward problem formulation with strong approximation power on large datasets, resulting in better performance over traditional approaches [208]. For multi-label classification in general, neural networks have achieved superior performance than previous well-established algorithms, e.g. adapted SVM, decision tree and boosting-based approaches, as reviewed in [209] and compared in [208]. The study [208] proposed BP-MLL, which is the first adaptation of the backpropagation algorithm in a feed-forward neural network for multi-label classification. BP-MLL has a new error function that optimise the difference within any pairs of outputs where one corresponding to correct labels and the other corresponding to incorrect labels. Later, the study [128] showed that cross-entropy loss function outperforms this error function in large-scale multi-label text classification. Recent studies on automated social annotation mostly adapt deep learning approaches, as in annotation of microblogs [67, 89, 110, 210] and academic papers [77]. Some of the notable neural network models adapted for multi-label classification are variations of Recurrent Neural Networks (RNN) [77, 89, 110] and Convolutional Neural Networks (CNN) [67, 210] with attention or memory mechanisms. The chapter above has been following this line of research on deep learning, and proposed a novel model for automated social annotation

based on several variants of Recurrent Neural Networks, such as Bidirectional Gated Recurrent Units (Bi-GRU) [35, 151] and Hierarchical Attention Networks (HAN) [200].

The use, representation and reasoning of knowledge has been a research frontier for deep learning based applications [68, p. 482-485]. Recent studies include Knowledge Graph Embedding approaches that represent concepts and relations in KBs [25] (as reviewed in [189]) and (attention-based) memory networks [24, 167] that help leveraging explicit facts in to deep learning models. The studies, however, mainly focused on leveraging knowledge as the input rather than the output. The important case in multi-label classification problems is that, knowledge particularly takes the form of label correlation in the output space. Few studies explored the leveraging of knowledge for deep learning adapted to multi-label classification problems. The work in [102] and [10] proposed a weight initialisation approach, which assigns a neuron in the penultimate layer of the neural network to "memorise" each pattern of label correlation. This can be inefficient (if not impossible) for huge amount of labels (user-generated tags) and their relation patterns in social annotation. The proposed model in the thesis provides a more feasible method, semantic-based loss regularisation, to leverage the structured knowledge for deep learning based multi-label classification.

Attention mechanisms have in recent years been a building block in deep learning models for natural language processing and also for social annotation [67, 77, 89, 110, 210]. Essentially, the original attention mechanisms in [9] are a soft *alignment* of each part (e.g. a word) in the data instance (e.g. a sentence) to a context so as to form an average-weighted representation to focus on the most important part in the data instance. The idea that attention mechanisms can learn to select the important parts in a sentence is applied to text classification, where the most representative work is Hierarchical Attention Network [200] which captures the hierarchical pattern of a document and treats each word or sentence distinctively for classification. Unlike in neural machine translation, there is no target representation that it can be aligned to (*cf.* [9, 117] and [200]), thus, a learnable vector was added and attended to each word or sentence. The idea of aligning each word or sentence to learnable vectors, although being used in later studies for sentiment classification [101] and document annotation [77], does not yet fully mimic the behavioural patterns of reading. Words and sentences can be alternatively guided by *explicit* metadata in the document, such as the title. Besides, while sentences are key elements for document understanding, recent studies in [111, 192] mainly model socially shared documents (answers in [111] and conversations in [192]) with word-level attention mechanisms. The *sentence-level*, *guided attention mechanisms* have been one key focus throughout this chapter.

Besides natural language processing, attention mechanisms have also been applied to computer vision, including image captioning [198] and multimodal image and text annotation [210]. The work in [198] models the presence of attention in human visual system for image captioning. The work in [210] models the mutual and external alignment

between texts and images in a microblog with a co-attention network for hashtag annotation. Distinct from [210], the proposed guided attention mechanisms in this chapter focus internally on the relations between metadata within a document, which naturally simulates users' reading behaviour for document annotation.

4.6 Summary and Discussion

Automated social annotation aims at predicting a set of tags from objects (e.g. documents) shared by users on social media platforms. The task can alleviate the incompleteness issue of social tagging data, and can then benefit the organisation, search and recommendation in many social media platforms. In this chapter, we have formulated the task as a multi-label classification problem and adapted a deep learning approach. We mainly tackled two problems: (i) how to leverage both similarity and subsumption relations among labels in neural networks to improve the performance of multi-label classification; and (ii) how to model users' reading and annotation behaviour, especially regarding the impact of the title metadata.

To leverage the structured knowledge of label correlation, which is a crucial issue for a high-dimensional label space (i.e. with large number of labels) [62, 209], we proposed two semantic-based loss regularisers which can enforce nodes in the output layer of a neural network to conform to the semantic relations, i.e., similarity and subsumption, among labels. The relations are acquired through inducing from the label sets and from grounding to external KBs, such as Microsoft Concept Graph (MCG). We applied this novel joint loss with the regularisers as a part of JMAN, and also on the Bi-GRU and HAN models. The results demonstrate consistent performance gain on the neural network models with the semantic-based loss regularisers.

To model the users' reading and annotation process, we designed a novel deep learning model, Joint Multi-label Attention Network (JMAN). Distinct from the previous Hierarchical Attention Network (HAN) [77, 200], JMAN separately encodes the title and the content and introduces a title-guided attention mechanism to align the title to each sentence. This design is according to previous studies on statistical analyses of users' annotation behaviour and the impact of the title metadata [54, 114]. Extensive experiments on four real-world datasets for social paper and question annotation show significant improvement of JMAN, in terms of accuracy, F_1 score and other metrics, over the popular, state-of-the-art baselines and model variations. In terms of F_1, JMAN significantly outperformed Bi-GRU (Bidirectional Gated Recurrent Unit) by relatively around 12.8% to 78.6%, and the Hierarchical Attention Network (HAN) by around 3.9% to 23.8%. A substantial reduction of training time was also achieved with the JMAN-s model, not applying the semantic-based loss regularisers. Analysis of the multi-source components showed the advantage of using the title-guided content representation and the proposed multiple sources in the document representation. The proposed title-guided sentence-level attention mechanism further improved the explainability over the HAN

model, by providing a new "view" on the sentence-level understanding, as analysed with the attention visualisation.

The overall approach has its assumptions and conditions. Theoretically, the approach is founded on the universal approximation theorem of neural networks [85, 149], so that a complex, non-linear function can be learned to match the document to labels. This assumption, however, is potentially compromised by real-world, noisy, high-dimensional datasets, especially when the size of the data is relatively low compared to a large number of labels for multi-label classification. This explains the overall low F_1 score in the performance (best F_1 score of 38.7% achieved for Bibsonomy and 26.0% for the CiteULike datasets). The semantic-based loss regularisers also assume the existence of label semantic relations to influence document multi-label classification. Also, the title-guided attention mechanism assumes that the title contains the essential, salient information of the document, which usually holds but may not always be the case in social media platforms. The title metadata may not be available in many social media platforms, thus more flexible, or generic, guided attention mechanisms can be considered in different scenarios. The multi-source architecture also assumes a particular (title-and-content) document structure that may not be directly applicable to other types of socially shared document. Understanding these assumptions and conditions can help better adapt the approach to other scenarios, for example, multi-modal social data which contains both texts and images [210], and socially shared questions and answers with code on Stack Overflow.

The proposed approach is also not without limitations. The improvement with the semantic-based loss regularisers on the deep learning models is still marginal, although consistent in all experimental settings. This may be explained by the process that the encoder and the document-label matching can indirectly model some of the label correlation. As a potential remedy, we showed a performance improvement (an absolute increase of F_1 score by 0.2%-1.2% over JMAN) through a dynamic update of the label semantic matrices, *Sim* and *Sub*. This "dynamic" version adds further, negative constraints on the output layer of the neural network and allows the label semantics to be more compatible to the dataset, with the cost of substantially increased memory requirement. Another potential direction to leverage structured knowledge in labels is using continuous representations of knowledge entities such as Knowledge Graph Embeddings [25, 189] with other types of deep learning architectures for multi-label classification, such as sequence-to-sequence (or sequence generation) models [192, 199] and autoencoders [186, 202]. Another limitation of the work in this chapter is related to the label quality issues. Although we applied a systematic tag cleaning process for the datasets, the label sets (cleaned tag sets) still suffer from missing, incomplete issues and some "noisy" labels in the data not expressing the topic of the document. A more robust multi-label document annotation, which can mitigate the missing and noisy label issues, would warrant further studies.

Despite the conditions and limitations of the proposed approach described above,

both the guided attention mechanisms and the semantic-based loss regularisers are generalisable to other tasks and models. For future studies, it is worth exploring other types of guided attention mechanisms where the title is not available, for example, in microblog annotation, the tweets may be guided with historical information such as past microblogs of the same user, or guided by external sources of different modalities, such as sensor data to annotate events. The usage of semantic-based loss regularisers is also to be validated with other types of semantic relations grounded to various KBs or induced from the datasets. The proposed model could also shed light on the open problem of extreme multi-label text classification [115], where there are hundreds of thousands or millions of possible labels and thus further requires scalability. Another important direction is to extend the current approach to deal with emerging new labels as discussed in [214]. Although we mainly focused on RNN-based models, which have been very commonly used especially for text understanding, it is also interesting to integrates the semantic-based loss regularisers and the guided attention mechanism with other neural network encoders, including Convolutional Neural Networks [98], the attention-based network Transformer [181] and transfer learning approaches, the Bidirectional Encoder Representations from Transformers (BERT) [45].

After the exploration of both the learning and leveraging of structured knowledge from social media data, in the next chapter, we will conclude the research, review the original research hypothesis and questions, and discuss potential topics for future studies.

Chapter 5

Conclusions and Future Work

Information becomes knowledge when you use an argument to draw conclusions from it. ... Knowledge becomes wisdom when it is integrated into your whole way of looking at things. – David Evans, Paul Gruba and Justin Zobel [53, p. 104]

The discovery, representation, and use of knowledge are fundamental to many applications in machine learning, and more broadly, for AI in general. More recently, the power of knowledge has attracted further interests of research communities, with the presence of massive user-generated data and the development of new approaches in machine learning. In the current Web, where massive data are created by users, it is time to explore the transformation of such unstructured, noisy user-generated data into more structured forms of knowledge, to support many semantic-based applications such as text classification, information retrieval, and recommendation. The probabilistic and neural network-based machine learning approaches provide further opportunities and challenges to learn and leverage structured knowledge from user-generated social media data.

In the previous chapters, we have explored from reviewing the idea of structured knowledge (in Chapter 2) to both learning and leveraging structured knowledge from social tagging data (in Chapter 3-4). This conclusion chapter summarises the thesis in Section 5.1, then reviews the aim of the study and addresses the formulated research questions from the introduction chapter in detail in Section 5.2. Furthermore, future studies regarding learning various types of dynamic structured knowledge, efficient ways of leveraging structured knowledge, end-to-end knowledge-centred learning, and extending to other types of user-generated data, have been discussed in Section 5.3.

5.1 Research Summary

We presented the main research contributions in Section 1.4 at the start of this thesis, and now after a view of the whole research, we come back again to the original point of this thesis for a more comprehensive and compact summarisation.

The thesis has aimed to learn useful structured knowledge from user-generated social media data. A knowledge-centred view has been considered, as illustrated in **Chapter 1**: knowledge bridges the gap between massive user-generated data to semantic-based applications. Without structured knowledge, user-generated data would have been dormant and less useful due to their unstructured characteristics. In this thesis, we mainly focused on a popular type of data across many social media platforms, social tagging data. The challenges to process social tagging data are their issues of noisiness, flatness, sparsity, incompleteness, which prevent their knowledge discovery and usage.

We reviewed the different types of structured knowledge and the relevant concepts in **Chapter 2**. In the first part of **Chapter 2**, we listed several key relevant concepts from the literature, including Knowledge Bases, Knowledge Graph, ontologies, concept hierarchies, semantic relations, etc., then we defined "structured knowledge" as an abstract term encompassing all types of the concepts above, highlighting the structuredness in organising and representing knowledge. We can observe a spectrum of structured knowledge from low semantics to high semantics. On the bottom of this spectrum, folksonomies were included as a potential source to form structured knowledge, as they provide low-semantics, but rich tag co-occurrence relations to learn more explicit, paradigmatic relations and more formal structured knowledge. Then we reviewed the different approaches to learn structured knowledge from folksonomies or social tagging data, with their limitations. The most challenging and unsolved issues to learn structured knowledge from tags are, therefore, the representation of the highly ambiguous meaning of tags and the quantification of their semantic relations to yield more accurate machine learning models. We further discovered that there were few studies on Knowledge Base Enrichment through social tagging data. Another, more empirical, aspect of structured knowledge is leveraging it to support semantic-based, machine learning applications. In the second part of **Chapter 2**, we reviewed studies on leveraging structured knowledge, mainly for automated social annotation, as a representative type of machine learning application. After a summarisation of the application, we identified three facets regarding the roles of structured knowledge in the studies on automated social annotation, knowledge as tag co-occurrence relations, knowledge in deep learning applications, and as label correlation in multi-label classification. Although many studies have attempted to incorporate knowledge into deep learning in general, few have explored the use of structured knowledge to address the label correlation issue.

Then, the thesis shifted to proposing a new machine learning system to learn structured knowledge from social tagging data in **Chapter 3**. The main idea was to learn to predict accurate relations with features generated from probabilistic topic modelling, founded on a formal set of assumptions. Once the machine learning models are trained and tested, tag concept hierarchies can be formed through a Hierarchy Generation Algorithm which predicts and organises tag concepts progressively from top to down into hierarchies for Knowledge Enrichment. Comprehensive evaluation studies were conducted on the large, academic social tagging dataset Bibsonomy and three data-driven or

human-engineered KBs, DBpedia, Microsoft Concept Graph, and the ACM Computing Classification System. We performed three evaluation strategies, namely, relation-level evaluation, ontology-level evaluation, and the novel, Knowledge Base Enrichment based evaluation. Evaluation results show that the proposed approach can generate high quality and meaningful hierarchies to enrich existing Knowledge Bases. The study provides empirical results on Knowledge Base Enrichment from user-generated social media data.

Regarding the leveraging of structured knowledge, the next part of the thesis explored and proposed a knowledge-enhanced deep learning model for automated social annotation, in **Chapter 4**. Semantic-based loss regularisation has been proposed to enhance the deep learning model with the similarity and subsumption relations between tags. Besides, to mimic the users' reading and annotation behaviour, a new form of attention mechanisms, guided attention mechanisms, have been proposed to guide the reading of sentences through the representation of the title metadata. The overall proposed deep learning model, Joint Multi-label Attention Networks (JMAN), can leverage the relations between tags, and separately models the title and the content of each document and injects an explicit, title-guided attention mechanism into each sentence. Extensive experiments on four datasets from real-world applications show a significant improvement of the JMAN model over state-of-the-art, popular baseline methods, with consistent performance gain of the semantic-based loss regularisers on deep learning models.

To recap the overall contribution, the study starts from a systematic, knowledge-centred view and then provides contributions on two aspects, learning structured knowledge and leveraging structured knowledge. To learn structured knowledge, the thesis proposed a machine learning systems founded on a set of assumptions based on probabilistic topic modelling, with comprehensive evaluation especially on the novel, Knowledge Based Enrichment based Evaluation. To leverage structured knowledge, semantic-based loss regularisers were proposed to constrain neural network models for multi-label classification, with a novel neural network model, JMAN, which incorporates a title-guided sentence-level attention mechanism to mimic the users' collective annotation behaviour. The research can shed light on both theoretical and empirical studies using probabilistic and deep learning based approach for knowledge engineering and computing with user-generated texts.

5.2 Research Findings

After the brief summary above, this section presents research findings in detail, regarding the investigation of the original hypothesis and the research questions formulated in Section 1.3.

The original hypothesis of this research was *substantial part of knowledge can be*

learned from user-generated social media data. We chose social tagging data as a representative type of social media data for their popularity and typical unstructured characteristics. According to the results of Knowledge Base Enrichment based evaluation in Section 3.6.4.3, it is clear most of the learned relations can enrich existing KBs. A large number of direct subsumption relations were generated and almost all of them (around 99%) were not present in the two existing KBs, DBpedia or CCS; in total, there were 3,846 distinct new relations for DBpedia and 1,302 for CCS. From the manual evaluation of a sample of the learned subsumption relations by domain experts, 67.64% of the ratings were either "subsumption" (38.44%) or "related" (29.20%), from four rating options (the other two options were "unrelated" and "unsure"). These results together show the novelty and correctness of the learned relations. The quality of the learned hierarchies was further validated through relation-level evaluation in Section 3.6.4.1, achieving 56.05% F_1 score, much better than previous approaches based on co-occurrence features (45.39%) and probabilistic topic modelling features (46.51%). Also, in ontology-level evaluation in Section 3.6.4.2, the learned hierarchies in different domains showed comparably and consistently better similarity to gold standard hierarchies than with previous approaches. Visualisation of learned hierarchies further validated the hypothesis. The hypothesis was further tested in terms of leveraging structured knowledge from tags to support automated social annotation. The structured knowledge of tag similarity and subsumption relations (integrated through semantic-based loss regularisation) provides consistent improvement of performance in automated social annotation, modelled with deep neural networks as a multi-label classification problem, as shown in the Section 4.4.4.2.

Based on this research hypothesis, five research questions were formulated in 1.3. We list the research questions below and provides research findings regarding each of them.

Research Question 1. *How to address the noisiness, ambiguity, sparsity, and incompleteness issues of social tagging data?*

This question is regarding the unstructured characteristics of all types of user-generated social media data. We identified these unstructured characteristics in Section 1.1 and reviewed them more clearly in Section 2.2.1. To address the noisiness issue and a part of ambiguity and sparsity issues, we proposed the Data Cleaning module in 3.2 to transform noisy tags into tag concepts. The effectiveness of this Data Cleaning module has been validated on three social tagging datasets, Bibsonomy, CiteULike-a, and CiteULike-t.

The ambiguity and sparsity issues of social tagging data were further addressed through data representation based on probabilistic topic modelling in Section 3.3. With probabilistic topic modelling, such as LDA, the ambiguous meaning of tag concepts can be softly and densely represented as an interpretable, low-dimensional vector. This probabilistic-based tag representation allows to further quantify the degree of subsumption between two tag concepts for the machine learning system.

The incompleteness issues of social tagging data were proposed to be addressed through automated social annotation, which is a typical semantic-based application to automatically suggest tags to newly shared or previously nontagged documents (i.e. micro-blogs, questions, papers, images, etc.) on social media platforms. We focused on textual documents such as papers (abstracts) shared in Bibsonomy and CiteULike, and questions asked in Zhihu. The proposed deep learning model, Joint Multi-label Attention Networks (JMAN), performs significantly better than the popular and state-of-the-art approaches in terms of the evaluation metrics, as shown in Section 4.4.4.1. The automated annotation results and the attention mechanisms were further qualitatively visualised in Section 4.4.8.

Thus, these unstructured issues were considered during the research process from learning structured knowledge to leveraging structured knowledge from social tagging data. The other issue, the flatness of tagging data, was addressed through learning relations and hierarchies, related to the next question.

Research Question 2. *How to learn subsumption relations and concept hierarchies from social tagging data?*

Subsumption relations and concept hierarchies are two major types of structured knowledge, defined in Section 2.1. Subsumption relations express the abstraction of concepts. Concept hierarchies represent hierarchies of a domain formed by subsumption relations.

The approaches to learning subsumption relations include heuristics-based, semantic grounding, unsupervised, and supervised approaches, as reviewed in Section 2.3. Based on the discussion of their limitations to learning useful hierarchies with explicit relations, a machine learning system was proposed and evaluated in Chapter 3. This binary classification model takes two ordered tag concepts with a context concept as input, and output whether a subsumption relation holds between them. Feature generation is the core part of this system, founded on three assumptions, namely, topic similarity, topic distribution, and probabilistic association. The main idea behind the assumptions is that subsumption relations can be quantified through latent topic information inferred with probabilistic topic models: topic similarity based features aim to ensure that certain similarity between concepts in terms of their topics; topic distribution features can quantify subsumption relations based on topic coverage and focus; and probabilistic association features are proposed to measure the association between concepts with conditional and joint probabilities. The combined three feature sets can be applied to detect subsumption relations between tag concepts.

Concept hierarchies can then be formed by organising the learned subsumption relations in a hierarchical manner. This was achieved through a Hierarchical Generation Algorithm in Section 3.5, which progressively predicts subsumption relations with the trained classification model from top to down and generates a concept hierarchy, starting from a user-specified root concept.

Research Question 3. *How to formally evaluate the learned structured knowledge from social tagging data?*

There have been few studies to formally evaluate the learned structured knowledge from social tagging data, as reviewed in Section 3.7. This thesis proposed a new set of evaluation strategies, relation-level evaluation, ontology-level evaluation, and Knowledge Base Enrichment based evaluation. Relation-level evaluation measures the performance on predicting (subsumption) relations; ontology-level evaluation measures the quality of the learned hierarchies through their resemblance to gold standard hierarchies, and Knowledge Base Enrichment based evaluation focuses on the qualitative and manual evaluation of the learned relations and hierarchies. Especially, as far as the author concerns, the Knowledge Base Enrichment based evaluation regarding the knowledge learned from folksonomies was not applied in previous literature. The three evaluation strategies together can assess more comprehensively the quality of the learned structured knowledge.

In the experiments, the academic social tagging data Bibsonomy was tested, with three KBs, DBpedia, Microsoft Concept Graph, and ACM Computing Classification System (CCS). Relation-level evaluation results demonstrated that the feature extraction mechanism based on probabilistic topic modeling outperforms, with a large margin of F_1, the mechanisms based on co-occurrence, and the proposed single feature sets to quality subsumption relations. The learned hierarchies with the proposed approach show the overall highest resemblance to the gold standard hierarchies, in the ontology-level evaluation. Further, the Knowledge Base Enrichment based evaluation show that the generated concept hierarchies can largely enrich the existing KBs, DBpedia and CCS.

Another aspect of evaluation is pragmatic or task-oriented evaluation, which assesses the learned knowledge through downstream applications, such as navigation as simulated in [165]. The thesis did not carry out a formal pragmatic evaluation of the learned structured knowledge, however, results from the automated social annotation in Chapter 4 showed that similarity and subsumption relations of tags can help improve the performance of deep learning based multi-label classification. This provides a new perspective to assess structured knowledge from social media data.

Research Question 4. *How to leverage structured knowledge to tackle the label correlation issue in deep learning based multi-label classification?*

Leveraging structured knowledge for machine learning applications has been long studied since the 2000s and earlier [21] (see Section 2.4), and can be traced back to knowledge-based systems in AI from the 1960s (as reviewed at the start of Chapter 1). However, recent advances in deep learning provide further challenges and opportunities to enhance machine learning models with knowledge. While most recent studies focus on embedding knowledge into vector spaces and as memories, input to the neural networks, much fewer studies explored knowledge in deep learning based multi-label classification.

In the latter case, structured knowledge is required in the output of neural networks, as reviewed in Section 2.4.3, 2.4.4, and also in 4.5.

In this thesis, we have shown that semantic-based loss regularisation is a viable approach to constrain the output of neural networks to the semantic relations of labels. For automated social annotation as a typical application, the labels are tag concepts cleaned from user-generated tags. The idea is to enforce the value of nodes (corresponding to the labels or tag concepts) in the layer after the sigmoid layer to conform to the semantic relations of labels. The similarity loss L_{sim} was designed to penalise the case of semantically similar and co-occurring labels in a label set with largely different predictions; the subsumption loss L_{sub} can further penalise the case that a child label is predicted as true, while the parent label is predicted as false, when they have a subsumption relation and co-occur in the label set of a document. This approach is extensible to many neural network models, including recurrent and convolutional neural networks with attention mechanisms, and is suitable for the case of high-dimensional label space. Experiments on attention-based neural network models, Bi-GRU, HAN, and the proposed JMAN show consistent improvement with the semantic-based loss regularisers, on four real-world datasets in socially paper and question annotation.

We also noted that the improvement, while consistent among the models and datasets, is mostly not significant. The absolute increase of F_1 score was higher with Bi-GRU (from 0.9% to 1.6%) and HAN (from 0.6% to 1.6%), and relatively lower with JMAN (from 0.1% to 0.5%), on the four datasets. This marginal improvement may be explained by the shared weights in the label-document matching process, which may already model much of the correlation among labels. While leveraging structured knowledge can lead to consistent improvement, it is worth to explore more efficient approaches to enhance neural networks with structured knowledge for multi-label classification. One potential approach towards this direction is the dynamic update of the matrices Sim and Sub, which allows the label semantics to be updated during training and to be more compatible to the dataset. The dynamic setting also adds further, negative constraints on the output layer through the negative values in Sim and Sub. The dynamic update of the label semantics (the JMAN$_d$ model) improved the performance over JMAN, i.e., the setting with fixed label semantics, absolutely by 0.2% to 1.2% in terms of F_1 on three of the four datasets, with the cost of substantial computational memory.

Research Question 5. *How to model users' social annotation process through deep learning?*

The last question focuses on modelling users' behaviour to improve the performance of automated social annotation. Previous studies modelled annotation behaviour based on features from tag co-occurrence and content, or applied matrix factorisation, graph-based approaches, and probabilistic graphical models, while recent studies applied deep learning models with attention mechanisms to model this process, and achieved superior performance, as reviewed in Sections 2.4.1 and 4.5. Deep learning models encode the

input texts as continuous vector representations, with recurrent or convolutional layers, and approximate the matching from the input to the label space.

It is important to identify the patterns of user behaviour before modelling. In this thesis, to enhance the current deep learning models, we focused on the reading behaviour regarding the impact of the title for annotation. Previous statistical analyses show that the title has a great impact on users' word choice in tagging [114] and also on document categorisation and tag recommendation [54]. This inspired the creation of title-guided attention mechanisms in the model. The proposed JMAN model separately inputs the title and sentences in the content and then uses the title to "guide" the reading of sentences, i.e. align to sentence representations. This explicit attention with the title representation is different from HAN, which aligns a learnable vector to the sentence representations. Compared to the word level in guided attention mechanisms, this focus on the sentence level was less explored in previous studies on modelling user-generated social texts [111, 192]. Experiments show that the title-guided sentence-level attention mechanism can further improve the performance of the model. In most evaluation settings, JMAN (with the "dynamic" version JMAN_d) and JMAN-s (the model without semantic-based loss regularisers) significantly outperformed the state-of-the-art, deep learning models, HAN and Bi-GRU, the downgraded models, JMAN-s-tg and JMAN-s-att (not using either the title-guided or the original sentence-level attention mechanisms), and the traditional approaches, SVM and LDA, in terms of the evaluation metrics. In terms of F_1, JMAN significantly outperformed Bi-GRU by relatively around 12.8% to 78.6%, and HAN by around 3.9% to 23.8%, on four real-world datasets. The training speed was substantially accelerated with the JMAN-s model, around 21.2%-54.7% faster than Bi-GRU and around 13.3%-23.2% faster than HAN on all datasets. Analysis on multi-source components further shows the effectiveness of the combined sources to model the annotation process. The efficiency of JMAN and its model variations can shed light on attention-based deep learning approaches to model users' reading and annotation behaviour.

Besides, users' collaborative tagging behaviour also inspired the design of the semantic-based loss regularisers (summarised in Research Question 4). Since different users may annotate the same shared document with tags (or labels) of different form and granularity, for automated social annotation, the structured knowledge of labels needs to be considered.

5.3 Future Studies

We previously discussed the assumptions, conditions, limitations, and directions warranting further studies in Section 3.8 and 4.6 (at the end of Chapters 3-4) regarding the proposed methods to learn and to leverage structured knowledge. This section

focuses on summarising broader areas for future studies, including various forms of dynamic structured knowledge, efficient ways of learning structured knowledge, end-to-end knowledge-centred learning, and extending to other types of user-generated data.

5.3.1 Learning Various Types of Structured Knowledge

Although the thesis has focused on learning the essential forms of structured knowledge, subsumption relations and concept hierarchies, user-generated social media data would convey knowledge of other types. Future studies could explore learning various forms of structured knowledge. A range of much wider, and more specific association relations [164], such as "is useful for", "located in", "derives from", etc., could be learned from social tags and other social media data. Formal ontologies, KGs, poly-hierarchies (where concepts can have multiple hypernym concepts) [194, p.140] may also be interested.

Besides, the *dynamic or temporal aspect* of structured knowledge worth being highlighted for future work, as most current studies have been mainly focusing on learning *static* knowledge. Structured knowledge needs evolving to respond to the change in real-world scenarios. One potential direction of future study is to adapt the current supervised learning method to an online learning framework to build evolving structured knowledge. The learned hierarchies could update itself with the availability of newly generated social media data. Also, the pattern of evolution of structured knowledge could be identified; this may be realised with recent approaches of dynamic word representation through deep learning [201] and probabilistic graphical models [147].

5.3.2 Efficient Approaches to Leverage Structured Knowledge

Although experiments show that the proposed semantic-based loss regularisers can leverage semantic relations of large sizes and can consistently boost various neural network models, the improvement so far is still marginal. One direction for future study is to explore more efficient approaches to leverage structured knowledge as label correlation for deep learning based multi-label classification. Recent studies proposed alternative neural network approaches for multi-label classification, for example, the work in the studies [186, 202] proposed novel *auto-encoder* architectures to input both instances and label sets to reconstruct label sets; the research in [192, 199] adapted *sequence-to-sequence networks* to generate each label set as a sequence. One of the advantages of these methods is that the label embedding or label representation could be jointly learned in the model to capture label correlation. So far, the studies modelled label correlation in an implicit manner, through a ranking loss between positive and negative labels [186, 202] or through the sequential dependence among output hidden states in sequence generation [199]. It is worth to further study the role of structured knowledge to improve deep learning approaches.

5.3.3 End-to-End Knowledge-Centred Learning

Deep learning approaches allow to use a single learning system to process complex tasks, without modular design and task-specific feature engineering; this approach of modeling is called "end-to-end" learning [64]. Although end-to-end learning is not with its limitations, it is shown in complex applications, such as natural language processing [38] and autonomous driving tasks [23], that this new learning paradigm leads to significantly better performance and smaller systems.

In this thesis, while a knowledge-centred framework was illustrated in Chapter 1, we separately modelled the learning and leveraging of structured knowledge for better explainability and easy adaptation to other related tasks. On the contrary, can we learn and leverage knowledge jointly using an end-to-end network? This would allow better optimisation of the learned knowledge for its usage in the final task. Recent advances in deep learning, especially attention mechanisms, transfer learning [146], knowledge graph embedding [25, 189], and graph neural networks [197], provide further means to represent and utilise structured knowledge. Future studies could explore end-to-end, knowledge-centred learning systems based on these advances.

5.3.4 Extending to Other User-Generated Data

While we mainly explored social tagging data, the proposed approach on learning and leveraging structured knowledge could be adapted to other types of user-generated data, such as microblogs, comments in e-business websites, and texts in Electrical Health Records (EHR). All these types of data share some common unstructured characteristics, and learning structured knowledge from them could be benefitted from the proposed methods in this thesis. On the other hand, the particularities of each type of data should be noted, for example the sentential and relatively richer contexts in microblogs and comments than social tags, and the variably structured, fragmented EHR data requiring domain specific knowledge [44]. Further studies need to test the proposed methods to learn and leverage structured knowledge from these types of data, taking into consideration of those data-specific factors.

5.4 Epilogue

The work reported in this thesis focused on two aspects of structured knowledge related to the massive amount of user-generated social media data: learning structured knowledge and leveraging structured knowledge. After a review of the background of structured knowledge, the "learning" part proposed a supervised learning system based on feature sets founded on probabilistic topic modelling to generate concept hierarchies from social tags; and the "leveraging" part modelled the users' social annotation with a knowledge-enhanced, attention-based deep learning model. The unstructured characteristics of noisiness, flatness, sparsity, and incompleteness of social tagging data had been considered during the design of the probabilistic-based and the neural network based

machine learning systems. Most relations in the learned hierarchies have shown to be able to enrich existing KBs. Structured knowledge such as similarity and subsumption relations consistently boost the performance of deep learning based multi-label classification. While the studies mainly focused on social tags, the proposed methods could be applied to various types of user-generated data. The results together demonstrated that a substantial amount of useful knowledge can be acquired through user-generated social media data. To recap, the work in this thesis provides a knowledge-centred view, bridging the gap between social media data and semantic-based applications with novel probabilistic-based and neural network based machine learning approaches, and shed lights on methods to efficiently learn and leverage various types of structured knowledge from user-generated data.

Appendix A

Visualisation of Tag Concept Hierarchies

The following figures present some selected tag concept hierarchies learned using the machine learning system proposed in Chapter 3 from the academic social tagging dataset, Bibsonomy. The domain of each concept hierarchy was specified by the author as an input to the Hierarchy Generation Algorithm.

FIGURE A.1: Excerpt of the learned hierarchy in the domain of *data mining*.

FIGURE A.2: Excerpt of the learned hierarchy in the domain of *social software*.

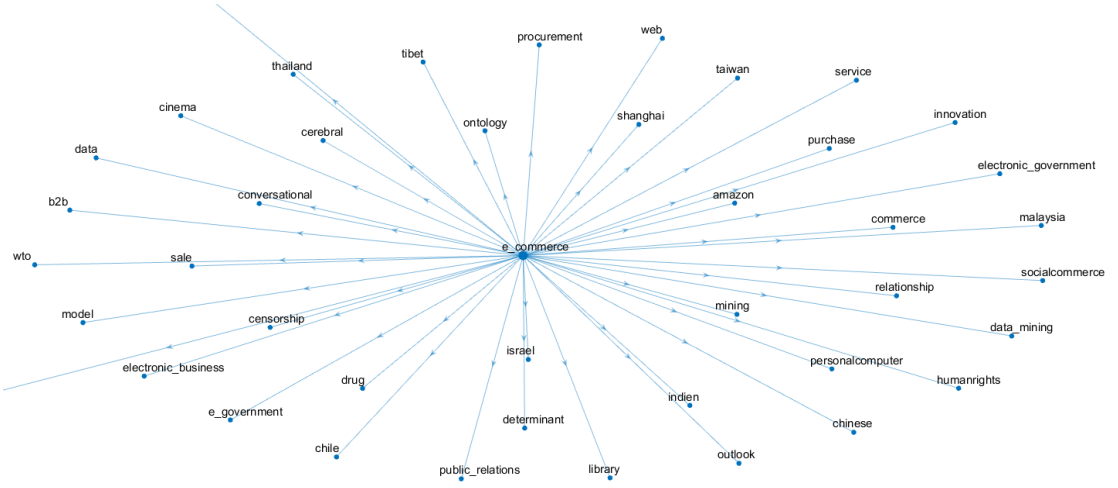


FIGURE A.3: Excerpt of the learned hierarchy in the domain of *e_commerce*.

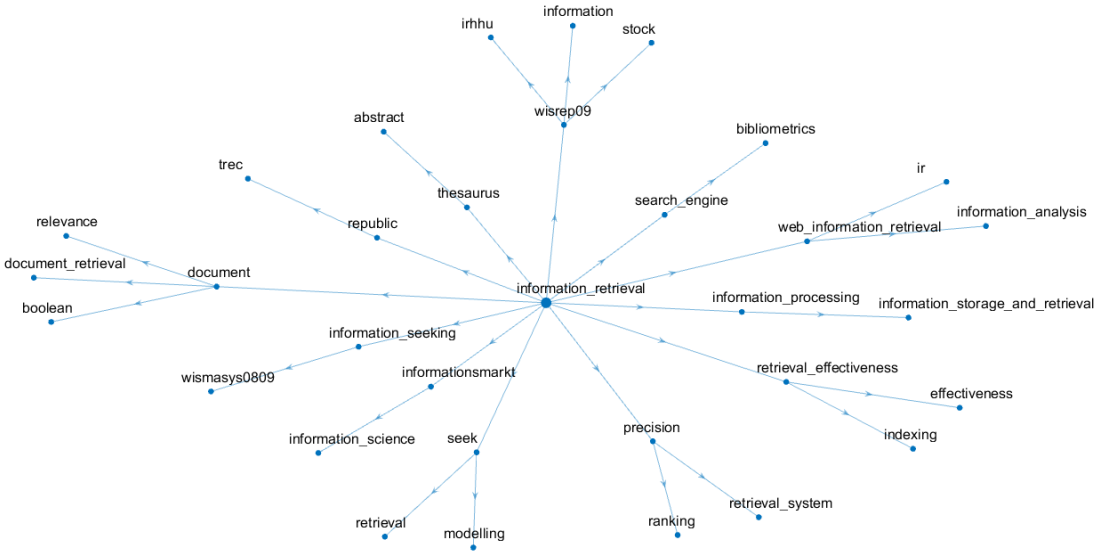


FIGURE A.4: Excerpt of the learned hierarchy in the domain of *information_retrieval*.

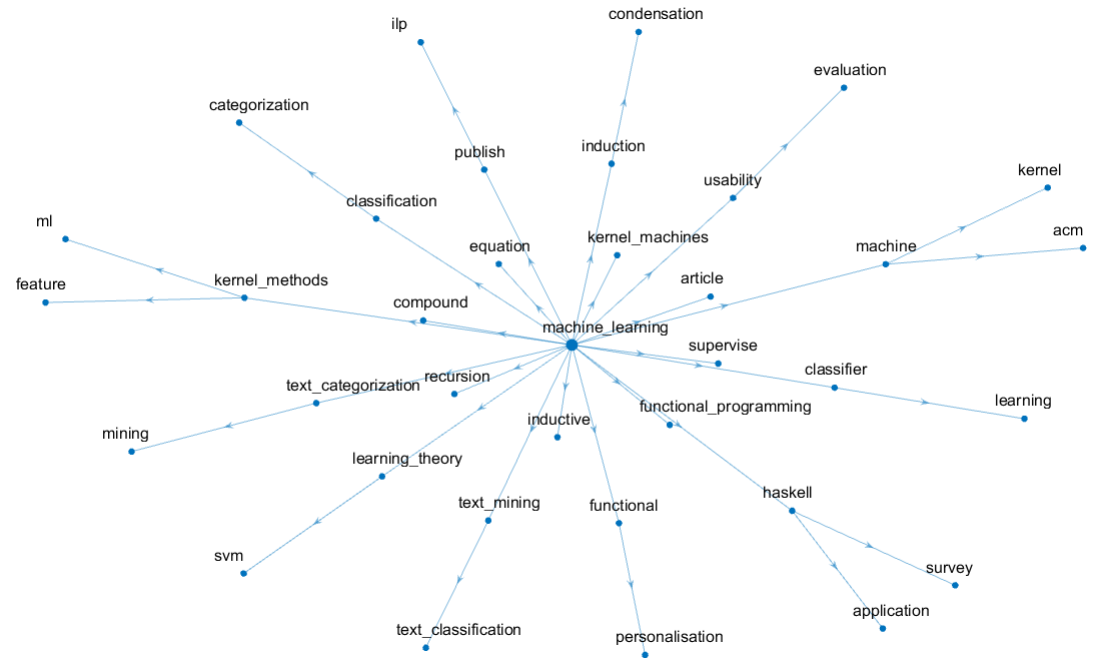


FIGURE A.5: Excerpt of the learned hierarchy in the domain of *machine_learning*.

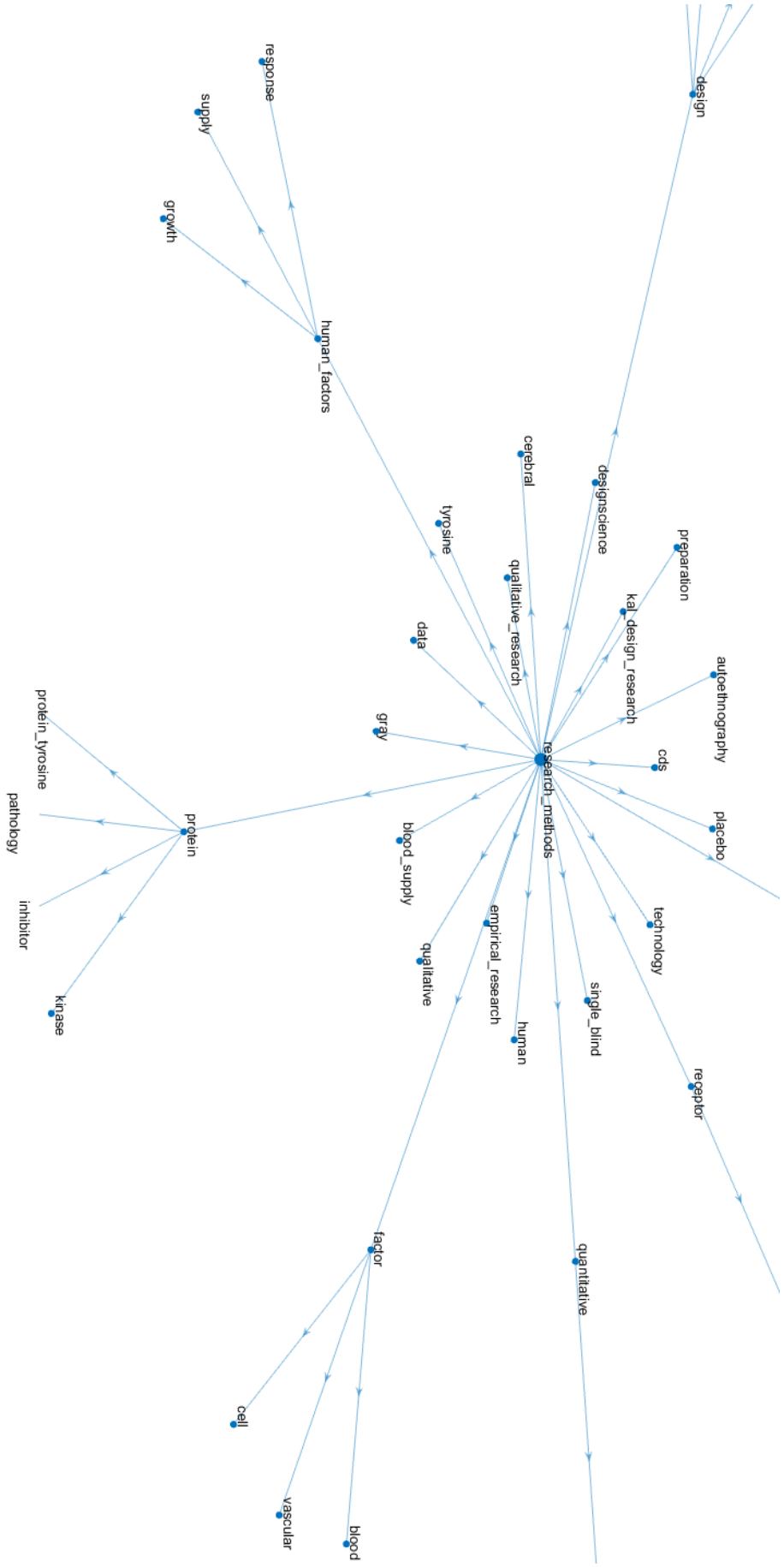


FIGURE A.6: Excerpt of the learned hierarchy in the domain of *research_methods*.

Appendix B

List of Open-Source Implementations

The implementation related to the research in this thesis has been documented in the following open-source projects on GitHub.

Tag-Data-Cleaning The project contains code and explanation of the steps in the Data Cleaning module in Section 3.2, and the preprocessed multiword and single tag groups (or tag concepts). `https://github.com/acadTags/Tag-Data-Cleaning`.

Tag-Relation-Learning The project contains implementation of Data Representation, Feature Generation, Classification and Testing, and Knowledge Enrichment modules of the machine learning system described in Chapter 3. The preprocessed datasets, Knowledge Bases, experimental results, and visualised hierarchies are also included. `https://github.com/acadTags/Tag-Relation-Learning`.

Automated-Social-Annotation The project contains implementation of the Joint Multi-label Attention Network (JMAN) and the baseline approaches, SVM, LDA, Bi-GRU, HAN, JMAN-s-att, JMAN-s-tg, JMAN-s, described in Chapter 4, with preprocessed datasets, Knowledge Bases, experimental results, and attention visualisation. `https://github.com/acadTags/Automated-Social-Annotation`.

Appendix C

Publications

Part of this thesis have been published or are currently under review:

- H. Dong, W. Wang, K. Huang, F. Coenen, *Automated Social Text Annotation with Joint Multi-Label Attention Networks*, Submitted to IEEE Transactions on Neural Networks and Learning Systems, 2020.
- H. Dong, W. Wang, F. Coenen, K. Huang, *Knowledge Base Enrichment by Relation Learning from Social Tagging Data*, Information Sciences, Volume 528, 2020, pp. 203-220.
- H. Dong, W. Wang, K. Huang, F. Coenen, *Joint Multi-Label Attention Networks for Social Text Annotation.* In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Volume 1 (Long and Short Papers), pp. 1348-1354.
- H. Dong, W. Wang, and F. Coenen, *Learning Relations from Social Tagging Data*, PRICAI 2018: Trends in Artificial Intelligence, 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28-31, Proceedings, Part I. Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, vol 11012. Springer, Cham, 2018, pp. 29-41.
- H. Dong, W. Wang, F. Coenen. *Rule for Inducing Hierarchies from Social Tagging Data*, in Chowdhury G., McLeod J., Gillet V., Willett P. (eds) Transforming Digital Worlds. iConference 2018, Sheffield, UK, 25-28 March. Lecture Notes in Computer Science, vol 10766. Springer, Cham, 2018, pp. 345-355.
- H. Dong, W. Wang, F. Coenen. *Deriving Dynamic Knowledge from Academic Social Tagging Data: A Novel Research Direction*, in iConference 2017 Proceedings, Wuhan, China, 22-25 March, 2017, pp. 661-666.
- H. Dong, W. Wang, and H.-N. Liang, *Learning Structured Knowledge from Social Tagging Data: A Critical Review of Methods and Techniques*, in 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), 8th

IEEE International Conference on Social Computing and Networking (IEEE SocialCom 2015), Chengdu, China, 19-21 December, 2015, pp. 307-314.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *Tensorflow: A system for large-scale machine learning*, Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (Berkeley, CA, USA), OSDI'16, USENIX Association, 2016, pp. 265–283.

[2] M.J. Adler and C. Van Doren, *How to read a book*, Touchstone, 2011.

[3] Fahad Almoqhim, David E. Millard, and Nigel Shadbolt, *Improving on popularity as a proxy for generality when building tag hierarchies from folksonomies*, Social Informatics: 6th Int. Conf. (Cham), Springer International Publishing, 2014, pp. 95–111.

[4] Mohammed Alruqimi and Noura Aknin, *Bridging the gap between the social and semantic web: Extracting domain-specific ontology from folksonomy*, Journal of King Saud University - Computer and Information Sciences **31** (2019), no. 1, 15 – 21.

[5] Hugo Alves and André Santanchè, *Folksonomized ontology and the 3e steps technique to support ontology evolvement*, Web Semantics: Science, Services and Agents on the World Wide Web **18** (2013), no. 1, 19 – 30.

[6] Pierre Andrews and Juan Pane, *Sense induction in folksonomies: a review*, Artificial Intelligence Review **40** (2013), no. 2, 147–174.

[7] Pierre Andrews, Juan Pane, and Ilya Zaihrayeu, *Semantic disambiguation in folksonomy: A case study*, Advanced Language Technologies for Digital Libraries: International Workshops on NLP4DL 2009, Viareggio, Italy, June 15, 2009 and AT4DL 2009, Trento, Italy, September 8, 2009 (Berlin, Heidelberg), Springer Berlin Heidelberg, 2011, pp. 114–134.

[8] Sofia Angeletou, *Semantic enrichment of folksonomy tagspaces*, The Semantic Web - ISWC 2008: 7th International Semantic Web Conference, ISWC 2008, Karlsruhe,

Germany, October 26-30, 2008. Proceedings (Berlin, Heidelberg), Springer Berlin Heidelberg, 2008, pp. 889–894.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, *Neural machine translation by jointly learning to align and translate*, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[10] Simon Baker and Anna Korhonen, *Initializing neural networks for hierarchical multi-label text classification*, BioNLP 2017 (2017), 307–315.

[11] Fabiano M Belém, Jussara M Almeida, and Marcos A Gonçalves, *A survey on tag recommendation methods*, Journal of the Association for Information Science and Technology **68** (2017), no. 4, 830–844.

[12] Fabiano M. Belém, Eder F. Martins, Jussara M. Almeida, and Marcos A. Gonçalves, *Personalized and object-centered tag recommendation methods for web 2.0 applications*, Information Processing & Management **50** (2014), no. 4, 524 – 553.

[13] Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, and Gerd Stumme, *The social bookmark and publication management system bibsonomy*, The VLDB Journal **19** (2010), no. 6, 849–875.

[14] Dominik Benz, Andreas Hotho, Gerd Stumme, and Stefan Sttzer, *Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge*, Proc. of the 2nd Web Science Conference (WebSci10), 2010, pp. 1–8.

[15] Mike Bergman, *An intrepid guide to ontologies*, `http://www.mkbergman.com/374/an-intrepid-guide-to-ontologies/`, 2007.

[16] C.M. Bishop, *Pattern recognition and machine learning*, Information Science and Statistics, Springer, 2006.

[17] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann, *Dbpedia - a crystallization point for the web of data*, Journal of Web Semantics **7** (2009), no. 3, 154 – 165, The Web of Data.

[18] David M. Blei, *Probabilistic topic models*, Commun. ACM **55** (2012), no. 4, 77–84.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan, *Latent dirichlet allocation*, Journal of machine Learning research **3** (2003), no. Jan, 993–1022.

[20] Stephan Bloehdorn, Philipp Cimiano, Andreas Hotho, and Steffen Staab, *An ontology-based framework for text mining*, LDV Forum, vol. 20, 2005, pp. 87–112.

[21] Stephan Bloehdorn and Andreas Hotho, *Ontologies for machine learning*, Handbook on Ontologies (Steffen Staab and Rudi Studer, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 637–661.

[22] Toine Bogers and Vivien Petras, *Supporting book search: A comprehensive comparison of tags vs. controlled vocabulary metadata*, Data and Information Management **1** (2017), no. 1, 17–34.

[23] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al., *End to end learning for self-driving cars*, arXiv preprint arXiv:1604.07316 (2016).

[24] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston, *Large-scale simple question answering with memory networks*, CoRR **abs/1506.02075** (2015).

[25] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko, *Translating embeddings for modeling multi-relational data*, Advances in neural information processing systems, 2013, pp. 2787–2795.

[26] Willem Nico Borst and W.N. Borst, *Construction of engineering ontologies for knowledge sharing and reuse*, Ph.D. thesis, University of Twente, Netherlands, 9 1997.

[27] Alexander Budanitsky and Graeme Hirst, *Evaluating wordnet-based measures of lexical semantic relatedness*, Comput. Linguist. **32** (2006), no. 1, 13–47.

[28] S. Cai, H. Sun, S. Gu, and Z. Ming, *Learning concept hierarchy from folksonomy*, 2011 Eighth Web Information Systems and Applications Conference, Oct 2011, pp. 47–51.

[29] Miquel Centelles, *Taxonomies for categorization and organization in web sites*, Hipertext.net (2005), no. 3.

[30] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology **2** (2011), 27:1–27:27.

[31] B. Chen, W. Li, Y. Zhang, and J. Hu, *Enhancing multi-label classification based on local label constraints and classifier chains*, 2016 International Joint Conference on Neural Networks (IJCNN), July 2016, pp. 1458–1463.

[32] Chong Chen and Pengcheng Luo, *Enhancing navigability: An algorithm for constructing tag trees*, Journal of Data and Information Science **2** (2017), no. 2, 56–75.

[33] Junpeng Chen, Shuai Feng, and Juan Liu, *Topic sense induction from social tags based on non-negative matrix factorization*, Information Sciences **280** (2014), 16–25.

[34] Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu, *Title-guided encoding for keyphrase generation*, The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019., 2019, pp. 6268–6275.

[35] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, *Learning phrase representations using rnn encoder–decoder for statistical machine translation*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[36] Carol E. B. Choksy, *8 steps to develop a taxonomy*, Information Management Journal **40** (2006), no. 6, 31–32,34–36,38–41 (English), Copyright - Copyright ARMA International Nov/Dec 2006; Document feature - Diagrams; Tables; Illustrations; Last updated - 2013-07-30; CODEN - IMAJF2; SubjectsTermNotLitGenreText - United States–US.

[37] Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker, *Ontology learning*, Handbook on Ontologies (Steffen Staab and Rudi Studer, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 245–267.

[38] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, *Natural language processing (almost) from scratch*, J. Mach. Learn. Res. **12** (2011), 2493–2537.

[39] Nello Cristianini and John Shawe-Taylor, *An introduction to support vector machines: And other kernel-based learning methods*, Cambridge University Press, New York, NY, USA, 2000.

[40] D. Alan Cruse, *Hyponymy and its varieties*, pp. 3–21, Springer Netherlands, Dordrecht, 2002.

[41] Dan McCreary, *Patterns of semantic integration: Riding the next wave*, `http://www.danmccreary.com/presentations/sem_int/sem_int.ppt`, 4 2006, Presentation Slides.

[42] Klaas Dellschaft and Steffen Staab, *Measuring the similiarity of concept hierarchies and its influence on the evaluation of learning procedures*, Master's thesis, University of Koblenz-Landau, 2005.

[43] ———, *On how to perform a gold standard based evaluation of ontology learning*, 5th Int. Semantic Web Conf., Springer Berlin Heidelberg, 2006, pp. 228–241.

[44] Spiros Denaxas, Arturo Gonzalez-Izquierdo, Kenan Direk, Natalie K Fitzpatrick, Ghazaleh Fatemifar, Amitava Banerjee, Richard JB Dobson, Laurence J Howe, Valerie Kuan, R Tom Lumbers, et al., *Uk phenomics platform for developing and validating electronic health record phenotypes: Caliber*, Journal of the American Medical Informatics Association **26** (2019), no. 12, 1545–1559.

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[46] Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang, *Learning topical translation model for microblog hashtag suggestion*, IJCAI, 2013, pp. 2078–2084.

[47] Endang Djuana, Yue Xu, and Yuefeng Li, *Constructing tag ontology from folksonomy based on wordnet*, Proceedings of the IADIS International Conference on Internet Technologies and Society 2011, International Association for Development of the Information Society (IADIS), 2011, pp. 1–8.

[48] H. Dong, W. Wang, and H. N. Liang, *Learning structured knowledge from social tagging data: A critical review of methods and techniques*, 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Dec 2015, pp. 307–314.

[49] Hang Dong, Wei Wang, and Frans Coenen, *Rules for inducing hierarchies from social tagging data*, Transforming Digital Worlds, Springer International Publishing, Cham, 2018, pp. 345–355.

[50] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang, *Knowledge vault: A web-scale approach to probabilistic knowledge fusion*, Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '14, ACM, 2014, pp. 601–610.

[51] H Du, SAMUEL KW Chu, and FLORENCE TY Lam, *Social bookmarking and tagging behavior: an empirical analysis on delicious and connotea*, Proceedings of the 2009 International Conference on Knowledge Management, 2009, pp. 1–11.

[52] Jeffrey L. Elman, *Finding structure in time*, Cognitive Science **14** (1990), no. 2, 179–211.

[53] D. Evans, P. Gruba, and J. Zobel, *How to write a better thesis*, SpringerLink : Bücher, Springer International Publishing, 2014.

[54] Flavio Figueiredo, Henrique Pinto, Fabiano Belm, Jussara Almeida, Marcos Gonçalves, David Fernandes, and Edleno Moura, *Assessing the quality of textual features in social media*, Information Processing & Management **49** (2013), no. 1, 222 – 247.

[55] Joseph L Fleiss, *Measuring nominal scale agreement among many raters.*, Psychological bulletin **76** (1971), no. 5, 378.

[56] Thomas M. J. Fruchterman and Edward M. Reingold, *Graph drawing by force-directed placement*, Software: Practice and Experience **21** (1991), no. 11, 1129–1164.

[57] Andrs Garca-Silva, Leyla Jael Garca-Castro, Alexander Garca, and Oscar Corcho, *Building domain ontologies out of folksonomies and linked data*, International Journal on Artificial Intelligence Tools **24** (2015), no. 02, 1540014:1–1540014:22.

[58] Andrés García-Silva, Oscar Corcho, Harith Alani, and Asunción Gómez-Pérez, *Review of the state of the art: discovering and associating semantics to tags in folksonomies*, The Knowledge Engineering Review **27** (2012), no. 1, 57–85.

[59] Andrés García-Silva, Leyla Jael García-Castro, Alexander García, and Oscar Corcho, *Social tags and linked data for ontology development: A case study in the financial domain*, Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), ACM, 2014, pp. 32:1–32:10.

[60] D. Gašević, A. Zouaq, C. Torniai, J. Jovanović, and M. Hatala, *An approach to folksonomy-based ontology maintenance for learning environments*, IEEE Transactions on Learning Technologies **4** (2011), no. 4, 301–314.

[61] Fatih Gedikli and Dietmar Jannach, *Recommender systems, semantic-based*, Encyclopedia of Social Network Analysis and Mining, Springer New York, New York, NY, 2014, pp. 1501–1510.

[62] Eva Gibaja and Sebastián Ventura, *A tutorial on multilabel learning*, ACM Computing Survey **47** (2015), no. 3, 52:1–52:38.

[63] Fausto Giunchiglia and Ilya Zaihrayeu, *Lightweight ontologies*, Encyclopedia of Database Systems, Springer US, Boston, MA, 2009, pp. 1613–1619.

[64] Tobias Glasmachers, *Limits of end-to-end learning*, arXiv preprint arXiv:1704.08305 (2017).

[65] Shantanu Godbole and Sunita Sarawagi, *Discriminative methods for multi-labeled classification*, Advances in Knowledge Discovery and Data Mining (Berlin, Heidelberg) (Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, eds.), Springer Berlin Heidelberg, 2004, pp. 22–30.

[66] Scott A. Golder and Bernardo A. Huberman, *Usage patterns of collaborative tagging systems*, Journal of Information Science **32** (2006), no. 2, 198–208.

[67] Yuyun Gong and Qi Zhang, *Hashtag recommendation using attention-based convolutional neural network.*, IJCAI, 2016, pp. 2782–2788.

[68] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, Adaptive Computation and Machine Learning series, MIT Press, 2016.

[69] Thomas L Griffiths and Mark Steyvers, *Prediction and semantic association*, Advances in neural information processing systems (2003), 11–18.

[70] Thomas L. Griffiths and Mark Steyvers, *Finding scientific topics*, Proceedings of the National Academy of Sciences **101** (2004), no. suppl 1, 5228–5235.

[71] Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum, *Topics in semantic representation.*, Psychological review **114** (2007), no. 2, 211.

[72] Stephan Grimm, Andreas Abecker, Johanna Völker, and Rudi Studer, *Ontologies and the semantic web*, Handbook of Semantic Web Technologies (John Domingue, Dieter Fensel, and James A. Hendler, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 507–579.

[73] Thomas R. Gruber, *A translation approach to portable ontology specifications*, Knowledge Acquisition **5** (1993), no. 2, 199 – 220.

[74] Tom Gruber, *Every ontology is a treatya social agreementamong people with some common motive in sharing*, Interview by Dr. Miltiadis D. Lytras, Official Quarterly Bulletin of AIS Special Interest Group on Semantic Web and Information Systems **1** (2004), no. 3.

[75] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, *The weka data mining software: An update*, SIGKDD Explor. Newsl. **11** (2009), no. 1, 10C18.

[76] James A Hampton, *The combination of prototype concepts*, The psychology of word meanings (1991), 91–116.

[77] Hebatallah A. Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, and Alessandro Micarelli, *Semantic-based tag recommendation in scientific bookmarking systems*, Proceedings of the 12th ACM Conference on Recommender Systems (New York, NY, USA), RecSys '18, ACM, 2018, pp. 465–469.

[78] Marti A. Hearst, *Automatic acquisition of hyponyms from large text corpora*, Proceedings of the 14th Conference on Computational Linguistics - Volume 2 (Stroudsburg, PA, USA), COLING '92, Association for Computational Linguistics, 1992, pp. 539–545.

[79] Gregor Heinrich, *Parameter estimation for text analysis*, University of Leipzig, Tech. Rep (2008).

[80] Paul Heymann and Hector Garcia-Molina, *Collaborative creation of communal hierarchical taxonomies in social tagging systems*, Tech. report, Stanford, 2006.

[81] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina, *Social tag prediction*, Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '08, ACM, 2008, pp. 531–538.

[82] C. A. R. Hoare, *Algorithm 64: Quicksort*, Commun. ACM **4** (1961), no. 7, 321–.

[83] Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural computation **9** (1997), no. 8, 1735–1780.

[84] Gail Hodge, *Systems of knowledge organization for digital libraries: Beyond traditional authority files.*, ERIC, 2000.

[85] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al., *Multilayer feedforward networks are universal approximators.*, Neural networks **2** (1989), no. 5, 359–366.

[86] A. Hotho, *Wordnet improves text document clustering*, Proc. SIGIR 2003 Semantic Web Workshop, 2003.

[87] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, *A practical guide to support vector classification*, Tech. report, Dept. of Computer Science, National Taiwan University, 2003.

[88] ———, *A practical guide to support vector classification*, Tech. report, Dept. of Computer Science, National Taiwan University, 2003.

[89] Haoran Huang, Qi Zhang, Yeyun Gong, and Xuanjing Huang, *Hashtag recommendation using end-to-end memory networks with hierarchical attention*, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 943–952.

[90] Fouzia Jabeen and Shah Khusro, *Quality-protected folksonomy maintenance approaches: a brief survey*, The Knowledge Engineering Review **30** (2015), no. 5, 521C544.

[91] Sarthak Jain and Byron C Wallace, *Attention is not explanation*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 3543–3556.

[92] Arash Joorabchi, Michael English, and Abdulhussain E. Mahdi, *Automatic mapping of user tags to wikipedia concepts: The case of a q&a website c stackoverflow*, Journal of Information Science **41** (2015), no. 5, 570–583.

[93] Dan Jurafsky and James H. Martin, *Chapter 2: Regular expressions, text normalization, edit distance*, Third Edition Draft, 2018.

[94] ———, *Chapter 6: Vector semantics*, Third Edition Draft, 2018.

[95] Christine Keller, *Theoretical and practical perspectives on ontology learning from folksonomies*, Ph.D. thesis, Universität Stuttgart, 2010.

[96] H. Keller, J.A. Macy, and A. Sullivan, *The story of my life*, Grosset & Dunlap, 1905.

[97] Elham Khabiri, James Caverlee, and Krishna Y. Kamath, *Predicting semantic annotations on the real-time web*, Proceedings of the 23rd ACM Conference on Hypertext and Social Media (New York, NY, USA), HT '12, ACM, 2012, pp. 219–228.

[98] Yoon Kim, *Convolutional neural networks for sentence classification*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[99] Diederik P Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980 (2014).

[100] S. Kullback and R. A. Leibler, *On information and sufficiency*, Ann. Math. Statist. **22** (1951), no. 1, 79–86.

[101] Abhishek Kumar, Daisuke Kawahara, and Sadao Kurohashi, *Knowledge-enriched two-layered attention network for sentiment analysis*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), vol. 2, 2018, pp. 253–258.

[102] Gakuto Kurata, Bing Xiang, and Bowen Zhou, *Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence*, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 521–526.

[103] Raymond Y. K. Lau, J. Leon Zhao, Wenping Zhang, Yi Cai, and Eric W. T. Ngai, *Learning context-sensitive domain ontologies from folksonomies: A cognitively motivated method*, INFORMS Journal on Computing **27** (2015), no. 3, 561–578.

[104] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, *Deep learning*, Nature **521** (2015), 436.

[105] Danielle H. Lee and Titus Schleyer, *Social tagging is no substitute for controlled indexing: A comparison of medical subject headings and citeulike tags assigned to 231,388 papers*, Journal of the American Society for Information Science and Technology **63** (2012), no. 9, 1747–1757.

[106] F.W. Lehmann and E.Y. Rodin, *Semantic networks in artificial intelligence*, International series in modern applied mathematics and computer science, no. v. 2, Pergamon Press, 1992.

[107] D.B. Lenat and R.V. Guha, *Building large knowledge-based systems: representation and inference in the cyc project*, Addison-Wesley Pub. Co., 1989.

[108] Leo Obrst, *The ontology spectrum & semantic models*, `http://ontolog.cim3.net/file/resource/presentation/LeoObrst_20060112/OntologySpectrumSemanticModels--LeoObrst_20060112.ppt`, 1 2006, ONTOLOG collaborative work environment - Historic Archives: Presentation Slides in ConferenceCall 2006 01 12.

[109] Vladimir I Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals*, Soviet physics doklady, vol. 10, 1966, pp. 707–710.

[110] Yang Li, Ting Liu, Jing Jiang, and Liang Zhang, *Hashtag recommendation with topical attention-based lstm*, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3019–3029.

[111] Di Liang, Fubao Zhang, Weidong Zhang, Qi Zhang, Jinlan Fu, Minlong Peng, Tao Gui, and Xuanjing Huang, *Adaptive multi-attention network incorporating answer information for duplicate question detection*, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR19, Association for Computing Machinery, 2019, p. 95C104.

[112] Jiaqing Liang, Yi Zhang, Yanghua Xiao, Haixun Wang, Wei Wang, and Pinpin Zhu, *On the transitivity of hypernym-hyponym relations in data-driven lexical taxonomies.*, AAAI, 2017, pp. 1185–1191.

[113] R.K. Lindsay, *Applications of artificial intelligence for organic chemistry: The dendral project*, McGraw-Hill advanced computer science series, McGraw-Hill, 1980.

[114] Marek Lipczak and Evangelos Milios, *The impact of resource title on tags in collaborative tagging systems*, Proceedings of the 21st ACM conference on Hypertext and hypermedia (New York, NY, USA), ACM, 2010, pp. 179–188.

[115] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang, *Deep learning for extreme multi-label text classification*, Proceedings of the 40th International

ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR 17, Association for Computing Machinery, 2017, p. 115C124.

[116] Caimei Lu, Jung ran Park, and Xiaohua Hu, *User tags versus expert-assigned subject terms: A comparison of librarything tags and library of congress subject headings*, Journal of Information Science **36** (2010), no. 6, 763–779.

[117] Thang Luong, Hieu Pham, and Christopher D Manning, *Effective approaches to attention-based neural machine translation*, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.

[118] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung, *Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems*, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1468–1478.

[119] Alexander Maedche and Steffen Staab, *Measuring similarity between ontologies*, 13th Int. Conf. EKAW 2002 Proc., Springer Berlin Heidelberg, 2002, pp. 251–263.

[120] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to information retrieval*, Cambridge University Press, New York, NY, USA, 2008.

[121] Andrew Kachites McCallum, *Mallet: A machine learning for language toolkit*, http://mallet.cs.umass.edu, 2002.

[122] Deborah L McGuinness, *Ontologies come of age*, Spinning the semantic web: bringing the World Wide Web to its full potential (2002), 171–194.

[123] Pasquale De Meo, Giovanni Quattrone, and Domenico Ursino, *Exploitation of semantic relationships and hierarchical data structures to support a user in his annotation and browsing activities in folksonomies*, Information Systems **34** (2009), no. 6, 511–535.

[124] Peter Mika, *Ontologies are us: A unified model of social networks and semantics*, Web Semantics: Science, Services and Agents on the World Wide Web **5** (2007), no. 1, 5–15.

[125] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

[126] David R Millen and Jonathan Feinberg, *Using social tagging to improve social navigation*, Workshop on the Social Navigation and Community based Adaptation Technologies, Citeseer, 2006.

[127] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*, Adaptive Computation and Machine Learning series, MIT Press, 2018.

[128] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz, *Large-scale multi-label text classification — revisiting neural networks*, Machine Learning and Knowledge Discovery in Databases (Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 437–452.

[129] Douglas L. Nelson, Cathy L. McEvoy, and Thomas A. Schreiber, *The university of south florida free association, rhyme, and word fragment norms*, Behavior Research Methods, Instruments, & Computers **36** (2004), no. 3, 402–407.

[130] Liqiang Nie, Yi-Liang Zhao, Xiangyu Wang, Jialie Shen, and Tat-Seng Chua, *Learning to recommend descriptive tags for questions in social forums*, ACM Trans. Inf. Syst. **32** (2014), no. 1, 5:1–5:23.

[131] J. Paul, *The literary works of leonardo da vinci, compiled and edited from the original manuscripts*, The Literary Works of Leonardo Da Vinci, Compiled and Edited from the Original Manuscripts, 1883.

[132] Jeffrey Pennington, Richard Socher, and Christopher Manning, *Glove: Global vectors for word representation*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[133] I. Peters and P. Becker, *Folksonomies: Indexing and retrieval in web 2.0*, De Gruyter/Saur, 2009.

[134] Isabella Peters, *Folksonomies. indexing and retrieval in web 2.0*, 1st ed., Walter de Gruyter & Co., USA, 2009.

[135] Isabella Peters and Wolfgang G. Stock, *Folksonomy and information retrieval*, Proceedings of the American Society for Information Science and Technology **44** (2007), no. 1, 1–28.

[136] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, *Deep contextualized word representations*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 2227–2237.

[137] Robert M. Pirsig, *Zen and the art of motorcycle maintenance : an inquiry into values*, Morrow, New York, 1974.

[138] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang, *Correlative multi-label video annotation*, Proceedings of the 15th ACM International Conference on Multimedia (New York, NY, USA), MM '07, ACM, 2007, pp. 17–26.

[139] C. Ramisch, *Multiword expressions acquisition: A generic and open framework*, Theory and Applications of Natural Language Processing, Springer International Publishing, 2014.

[140] Justus J Randolph, *Free-marginal multirater kappa (multirater κfree): An alternative to fleiss fixed-marginal multirater kappa*, the Joensuu Learning and Instruction Symposium, 2005.

[141] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank, *Classifier chains for multi-label classification*, Machine Learning and Knowledge Discovery in Databases (Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 254–269.

[142] ———, *Classifier chains for multi-label classification*, Machine learning **85** (2011), no. 3, 333–359.

[143] Jesse Read, Peter Reutemann, Bernhard Pfahringer, and Geoff Holmes, *Meka: A multi-label/multi-target extension to weka*, J. Mach. Learn. Res. **17** (2016), no. 1, 667C671.

[144] Alex Sandro C. Rêgo, Leandro Balby Marinho, and Carlos Eduardo S. Pires, *A supervised learning approach to detect subsumption relations between tags in folksonomies*, Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15), ACM, 2015, pp. 409–415.

[145] Radim Řehůřek and Petr Sojka, *Software Framework for Topic Modelling with Large Corpora*, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks (Valletta, Malta), ELRA, May 2010, pp. 45–50 (English).

[146] Sebastian Ruder, *Neural transfer learning for natural language processing*, Ph.D. thesis, National University of Ireland, Galway, 2019.

[147] Maja Rudolph and David Blei, *Dynamic embeddings for language evolution*, Proceedings of the 2018 World Wide Web Conference (Republic and Canton of Geneva, Switzerland), WWW '18, International World Wide Web Conferences Steering Committee, 2018, pp. 1003–1011.

[148] S. Russell and P. Norvig, *Artificial intelligence: A modern approach*, Always learning, Pearson, 2016.

[149] Anton Maximilian Schäfer and Hans-Georg Zimmermann, *Recurrent neural networks are universal approximators*, International journal of neural systems **17** (2007), no. 04, 253–263.

[150] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme, *Mining association rules in folksonomies*, Data Science and Classification, Springer, 2006, pp. 261–270.

[151] Mike Schuster and Kuldip K Paliwal, *Bidirectional recurrent neural networks*, IEEE Transactions on Signal Processing **45** (1997), no. 11, 2673–2681.

[152] Hinrich Schütze and Jan Pedersen, *A vector model for syntagmatic and paradigmatic relatedness*, Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research, Oxford, 1993, pp. 104–113.

[153] Shilad Sen, F. Maxwell Harper, Adam LaPitz, and John Riedl, *The quest for quality tags*, Proceedings of the 2007 International ACM Conference on Supporting Group Work (New York, NY, USA), GROUP '07, ACM, 2007, pp. 361–370.

[154] Philipp Singer, Thomas Niebler, Andreas Hotho, and Markus Strohmaier, *Folksonomies*, Encyclopedia of Social Network Analysis and Mining, Springer New York, New York, NY, 2014, pp. 542–547.

[155] Amit Singhal, *Introducing the knowledge graph: things, not strings*, `https://www.blog.google/products/search/introducing-knowledge-graph-things-not/`, 2012, Google Official Blog, Cross-posted on the Google Inside Search Blog.

[156] Barry Smith and Christopher Welty, *FOIS introduction: Ontology—towards a new synthesis*, Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001 (New York, NY, USA), FOIS '01, ACM, 2001, Conference Chair-Smith, Barry and Conference Chair-Welty, Christopher, pp. .3–.9.

[157] Yang Song, Lu Zhang, and C. Lee Giles, *Automatic tag recommendation algorithms for social recommender systems*, ACM Trans. Web **5** (2011), no. 1, 4:1–4:31.

[158] Mohammad S Sorower, *A literature survey on algorithms for multi-label learning*, Tech. report, Department of Computer Science, Oregon State University, 2010.

[159] Renato Rocha Souza, Douglas Tudhope, and Maurício Barcellos Almeida, *Towards a taxonomy of kos: Dimensions for classifying knowledge organization systems*, KO KNOWLEDGE ORGANIZATION **39** (2012), no. 3, 179–192.

[160] Lucia Specia and Enrico Motta, *Integrating folksonomies with the semantic web*, The Semantic Web: Research and Applications (Berlin, Heidelberg) (Enrico Franconi, Michael Kifer, and Wolfgang May, eds.), Springer Berlin Heidelberg, 2007, pp. 624–639.

[161] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, The Journal of Machine Learning Research **15** (2014), no. 1, 1929–1958.

[162] starays, *Zhihu Machine Learning Challenge 2017 is over. Who won? What will Zhihu do in the future?*, `https://zhuanlan.zhihu.com/p/28912353`, 2017, Zhihu Technical Blog, Zhihu.

[163] Mark Steyvers and Tom Griffiths, *Probabilistic topic models*, Handbook of latent semantic analysis **427** (2007), no. 7, 424–440.

[164] Wolfgang G. Stock, *Concepts and semantic relations in information science*, Journal of the American Society for Information Science and Technology **61** (2010), no. 10, 1951–1969.

[165] Markus Strohmaier, Denis Helic, Dominik Benz, Christian Körner, and Roman Kern, *Evaluation of folksonomy induction algorithms*, ACM Trans. Intell. Syst. Technol. **3** (2012), no. 4, 74:1–74:22.

[166] Rudi Studer, V.Richard Benjamins, and Dieter Fensel, *Knowledge engineering: Principles and methods*, Data & Knowledge Engineering **25** (1998), no. 1, 161 – 197.

[167] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., *End-to-end memory networks*, Advances in neural information processing systems, 2015, pp. 2440–2448.

[168] Sue Yeon Syn and Michael B Spring, *Finding subject terms for classificatory metadata from user-generated social tags*, Journal of the Association for Information Science and Technology **64** (2013), no. 5, 964–980.

[169] P. Szymański and T. Kajdanowicz, *A scikit-based Python environment for performing multi-label classification*, ArXiv e-prints (2017).

[170] Farbound Tai and Hsuan-Tien Lin, *Multilabel classification with principal label space transformation*, Neural Computation **24** (2012), no. 9, 2508–2542.

[171] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, *Introduction to data mining, (first edition)*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[172] P.N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to data mining*, second ed., Pearson, 2019.

[173] James W Tanaka and Marjorie Taylor, *Object categories and expertise: Is the basic level in the eye of the beholder?*, Cognitive Psychology **23** (1991), no. 3, 457 – 482.

[174] Jie Tang, Ho-fung Leung, Qiong Luo, Dewei Chen, and Jibin Gong, *Towards ontology learning from folksonomies*, Proceedings of the 21st International Jont Conference on Artifical Intelligence (San Francisco, CA, USA), IJCAI'09, Morgan Kaufmann Publishers Inc., 2009, pp. 2089–2094.

[175] Q. T. Tho, S. C. Hui, and A. C. M. Fong and, *Automatic fuzzy ontology generation for semantic web*, IEEE Transactions on Knowledge and Data Engineering **18** (2006), no. 6, 842–856.

[176] Grigorios Tsoumakas and Ioannis Katakis, *Multi-label classification: An overview*, International Journal of Data Warehousing and Mining (IJDWM) **3** (2007), no. 3, 1–13.

[177] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas, *Effective and efficient multilabel classification in domains with large number of labels*, Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08), vol. 21, sn, 2008, pp. 53–59.

[178] ———, *Mining multi-label data*, Data Mining and Knowl. Discovery Handbook (Boston, MA) (Oded Maimon and Lior Rokach, eds.), Springer US, 2010, pp. 667–685.

[179] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas, *Mulan: A java library for multi-label learning*, Journal of Machine Learning Research **12** (2011), 2411–2414.

[180] Céline Van Damme, Martin Hepp, and Katharina Siorpaes, *Folksontology: An integrated approach for turning folksonomies into ontologies*, Bridging the Gap between Semantic Web and Web **2** (2007), no. 2, 57–70.

[181] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[182] B.M. Villazón-Terrazas, *A method for reusing and re-engineering non-ontological resources for building ontologies*, Ingenieria de las telecomunicaciones, IOS Press, 2012.

[183] Thomas Vander Wal, *Explaining and showing broad and narrow folksonomies*, `http://www.vanderwal.net/random/entrysel.php?blog=1635`, 2005, [Online; accessed 10-July-2018].

[184] ———, *Folksonomy*, `http://vanderwal.net/folksonomy.html`, 2007, [Online; accessed 10-July-2018].

[185] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno, *Evaluation methods for topic models*, Proceedings of the 26th Annual International Conference on Machine Learning (New York, NY, USA), ICML '09, ACM, 2009, pp. 1105–1112.

[186] Bingyu Wang, Li Chen, Wei Sun, Kechen Qin, Kefeng Li, and Hui Zhou, *Ranking-based autoencoder for extreme multi-label classification*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), 2019, pp. 2820–2830.

[187] Hao Wang, Binyi Chen, and Wu-Jun Li, *Collaborative topic regression with social regularization for tag recommendation*, Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13, AAAI Press, 2013, pp. 2719–2725.

[188] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo, *Dkn: Deep knowledge-aware network for news recommendation*, Proceedings of the 2018 World Wide Web Conference (Republic and Canton of Geneva, Switzerland), WWW '18, International World Wide Web Conferences Steering Committee, 2018, pp. 1835–1844.

[189] Q. Wang, Z. Mao, B. Wang, and L. Guo, *Knowledge graph embedding: A survey of approaches and applications*, IEEE Transactions on Knowledge and Data Engineering **29** (2017), no. 12, 2724–2743.

[190] Wei Wang, Payam Mamaani Barnaghi, and Andrzej Bargiela, *Probabilistic topic models for learning terminological ontologies*, IEEE Transactions on Knowledge and Data Engineering **22** (2010), no. 7, 1028–1040.

[191] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang, *Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach*, Proceedings of the 20th ACM International Conference on Information and Knowledge Management (New York, NY, USA), CIKM '11, ACM, 2011, pp. 1031–1040.

[192] Yue Wang, Jing Li, Irwin King, Michael R Lyu, and Shuming Shi, *Microblog hashtag generation via encoding conversation contexts*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 1624–1633.

[193] Jônatas Wehrmann, Rodrigo C. Barros, Silvia N. das Dôres, and Ricardo Cerri, *Hierarchical multi-label classification with chained neural networks*, Proceedings of the Symposium on Applied Computing (New York, NY, USA), SAC 17, Association for Computing Machinery, 2017, p. 790C795.

[194] Katrin Weller, *Knowledge representation in the social semantic web*, De Gruyter Saur, Berlin; New York, NY, 2010.

[195] Timothy Williamson, *On vagueness, or, when is a heap of sand not a heap of sand?*, `https://aeon.co/ideas/on-vagueness-when-is-a-heap-of-sand-not-a-heap-of-sand`, 2016, [Online; accessed 15-Sep-2019].

[196] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu, *Probase: A probabilistic taxonomy for text understanding*, Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (New York, NY, USA), SIGMOD '12, ACM, 2012, pp. 481–492.

[197] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu, *A comprehensive survey on graph neural networks*, arXiv preprint arXiv:1901.00596 (2019).

[198] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, *Show, attend and tell: Neural image caption generation with visual attention*, International conference on machine learning, 2015, pp. 2048–2057.

[199] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang, *SGM: sequence generation model for multi-label classification*, Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, 2018, pp. 3915–3926.

[200] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, *Hierarchical attention networks for document classification*, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[201] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong, *Dynamic word embeddings for evolving semantic discovery*, Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (New York, NY, USA), WSDM '18, ACM, 2018, pp. 673–681.

[202] Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang, *Learning deep latent space for multi-label classification*, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 2838–2844.

[203] Eva Zangerle, Wolfgang Gassler, and Gunther Specht, *Recommending#-tags in twitter*, Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings, vol. 730, 2011, pp. 67–78.

[204] Amrapali Zaveri, Dimitris Kontokostas, Mohamed A. Sherif, Lorenz Bühmann, Mohamed Morsey, Sören Auer, and Jens Lehmann, *User-driven quality evaluation of dbpedia*, Proceedings of the 9th International Conference on Semantic Systems (New York, NY, USA), I-SEMANTICS '13, ACM, 2013, pp. 97–104.

[205] Marcia Lei Zeng, *Knowledge organization systems (kos)*, KO KNOWLEDGE ORGANIZATION **35** (2008), no. 2-3, 160–182.

[206] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma, *Collaborative knowledge base embedding for recommender systems*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '16, ACM, 2016, pp. 353–362.

[207] Min-Ling Zhang and Kun Zhang, *Multi-label learning by exploiting label dependency*, Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '10, ACM, 2010, pp. 999–1008.

[208] Min-Ling Zhang and Zhi-Hua Zhou, *Multilabel neural networks with applications to functional genomics and text categorization*, IEEE Transactions on Knowledge and Data Engineering **18** (2006), no. 10, 1338–1351.

[209] Min-Ling Zhang and Zhi-Hua Zhou, *A review on multi-label learning algorithms*, IEEE Transactions on Knowledge and Data Engineering **26** (2014), no. 8, 1819–1837.

[210] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong, *Hashtag recommendation for multimodal microblog using co-attention network*, Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, AAAI Press, 2017, pp. 3420–3426.

[211] Mianwei Zhou, Shenghua Bao, Xian Wu, and Yong Yu, *An unsupervised model for exploring hierarchical semantics from social annotations*, The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings, Springer Berlin Heidelberg, 2007, pp. 680–693.

[212] Jiangang Zhu, Beijun Shen, Xuyang Cai, and Haofen Wang, *Building a large-scale software programming taxonomy from stackoverflow.*, SEKE, 2015, pp. 391–396.

[213] Y. Zhu, J. T. Kwok, and Z. Zhou, *Multi-label learning with global and local label correlation*, IEEE Transactions on Knowledge and Data Engineering **30** (2018), no. 6, 1081–1094.

[214] Y. Zhu, K. M. Ting, and Z. Zhou, *Multi-label learning with emerging new labels*, IEEE Transactions on Knowledge and Data Engineering **30** (2018), no. 10, 1901–1914.

[215] A. Zubiaga, V. Fresno, R. Martnez, and A. P. Garca-Plaza, *Harnessing folksonomies to produce a social classification of resources*, IEEE Transactions on Knowledge and Data Engineering **25** (2013), no. 8, 1801–1813.